IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 5, 2020, accepted November 4, 2020, date of publication November 16, 2020,
date of current version November 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038200

Extraction of Computer-Inherent Characteristics Based on Time Drift and CPU Core Temperature

NAOTO HOSHIKAWA[1], **RYO NAMIKI**[2], **KATSUMI HIRATA**[1], **(Member, IEEE), ATSUSHI SHIRAKI**[3,4], **AND TOMOYOSHI ITO**[3]

[1]Department of Innovative Electrical and Electronic Engineering, National Institute of Technology, Oyama College, Oyama 323-0806, Japan
[2]Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan
[3]Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan
[4]Institute of Management and Information Technologies, Chiba University, Chiba 263-8522, Japan

Corresponding author: Naoto Hoshikawa (hoshikawa.naoto@oyama-ct.ac.jp)

This work was supported by the Asahi Glass Foundation and Yazaki Memorial Foundation for Science and Technology.

ABSTRACT The number of computers that provide information services via the Internet is constantly increasing, particularly for IoT applications. Compared to the servers in managed data centers, IoT computers have an increased risk of contamination from unidentified computers. It is therefore important for applications that utilize IoT to identify the appropriate computers to use. However, it is difficult to assign digital identifiers with adequate protection to a huge number of IoT computers. In this work, we have devised a method to extract computer-specific features from the characteristics of the CPU core temperature and the drift of the time information. This feature data can be treated as computer-specific information, just like human biometric information. We performed experiments on two types of computer (three of each) with the same software settings and obtained a regression linear equation for each that represents the time deviation per temperature. The correlation coefficients of these equations were greater than 0.9 for all, and a strong positive correlation was obtained. From the equation and the temperature values, we found that it is possible to estimate the computer-specific time deviation. Our method does not require the implementation of a special temperature sensor. Therefore, it shows good potential for future applications.

INDEX TERMS Clocks, computer network management, Internet of Things, network service, parameter extraction, system identification, time measurement.

I. INTRODUCTION

The growing prevalence of the Internet of Things (IoT) has led to a dramatic increase in the number of computers connected to the Internet, which is predicted to exceed 50 billion in 2020 [1] and increase to 125 billion by 2030 [2]. IoT computers are a key infrastructure for running Industry 4.0 and Smart City applications [3]–[5]. However, since IoT computers are deployed in large numbers in various locations, it is more difficult to manage their quality compared to the servers in conventional tightly managed data centers. This increases the risk by contamination of unidentified computers and the spoofing of legitimate computers [6]. As such, a means of identifying legitimate computers for IoT applications is urgently needed.

Generally, network applications use identifier data such as Internet Protocol (IP) addresses, session IDs, application IDs, and user IDs, which are provided by a service or network administrator, to distinguish between computers. However, identifiers defined with digital data are easy to replicate and can be a source of spoofing. Various methods utilize biometric information such as fingerprints and irises to identify human beings in service authentication, and recently this technique has been applied to network services as well [7]. Unlike identifier data, biometric information has an advantage in that it is difficulty to falsify. Moreover, the lengthy process of issuing and assigning identifiers can be avoided. We have been investigating a method analogous to biometric authentication in which the unique characteristics of clock frequency signals generated by computer hardware are treated as identifier information.

Computers make use of various clock frequency signals, such as a microprocessor to drive the computer, a timer counter to manage program interrupts, and a real-time clock (RTC) to keep the time when the system is powered off. This signal is slightly different for each piece of hardware,

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai.

even if the same frequency value is set. In a prior work, we hypothesized that the relative signal deviation compared to the reference signal could be extracted as a computer-specific characteristic feature, which we dubbed a clock fingerprint (CFP) [8]. However, it is difficult to measure very high-frequency clock signals (e.g., computing processing units), and it is not practical to implement the hardware of a signal acquisition function for this specific purpose. Therefore, we devised a method that uses the time information and counter value instead of the clock signal frequency. Since computers generally get their time information from a clock counter that is generated from the clock frequency, this information is also subject to hardware-specific gaps. According to our hypothesis, we can analyze a computer's unique characteristics by examining the ''time drift'', which is the difference between the computer-generated time information and the reference time.

However, as the clock oscillator of a signal varies in frequency due to the effect of temperature [9], our previous method could not provide stable results in an environment where the temperature changes rapidly. In the present study, we developed a new feature extraction method corresponding to temperature change in which the hardware temperature and time drift characteristics are sampled to derive characteristic equations for each computer. In addition, since it is not practical to attach a temperature sensor to every clock oscillator, we devised a method to map the temperature of the clock oscillator from the central processing unit (CPU) core temperature information.

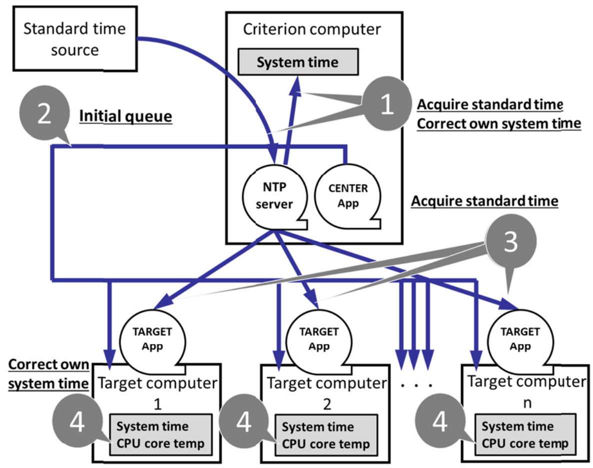
In this paper, we first present our method for obtaining characteristic feature values based on temperature and time drift for two types of computer with the same operating system (OS) and system settings. Next, we report the values obtained from experiments and the results of the linear functions derived from them. Finally, we discuss how these results validate our hypothesis, clarify the contributions of this work compared with related research, and briefly touch on future challenges.

II. METHODS

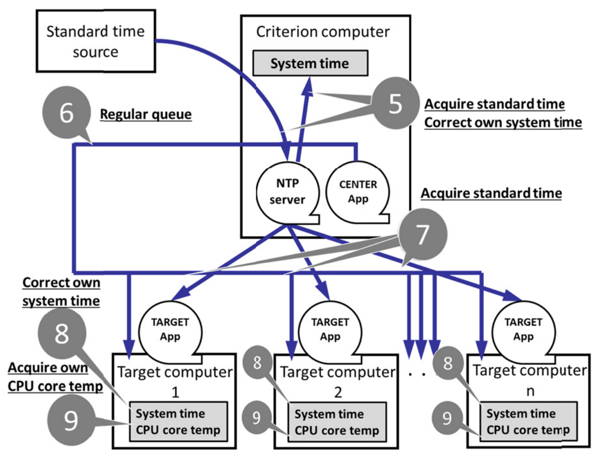
We define ''target computer'' as the computer from which the time drift features are extracted, ''criterion computer'' as the central computer that keeps the standard time and extracts the features, and ''standard time source'' as the reference source that provides the exact standard time. In the following, we explain the basic principle and the experimental system for deriving the characteristic equations with the temperature parameters proposed in this paper.

A. PRINCIPLE

The system time is the time information used by the operating system and applications. It is updated by adding up the values of the clock counter function on the computer. The counter source derived from the clock oscillator available to the computer depends on the computer type. For example, in the case of the Intel architecture, which is typically used in personal computers (PC), the system time is generated by calculating the CPU's clock counter value, called the time stamp counter (TSC). In any case, if the hardware that generates the counter values exhibits an inherent characteristic, we should be able to observe the inherent deviation as a time drift. Furthermore, if the clock frequency of the oscillator changes with temperature, the functional expression representing the time drift value with this temperature parameter can represent a characteristic equation of the target computer.



(a) Initial phase



(b) Repeat phase

FIGURE 1. Overview of feature extraction system and its processing flow.

The procedure for extracting features from the deviation of the system time of the target computer using the criterion computer is shown in Fig. 1 and described below. The criterion computer utilizes the functions of a feature data collection application and a Network Time Protocol (NTP) server. This collection application is hereafter referred to as the CENTER App. The target computers set up a feature data collection target application, which hereinafter is referred to as the TARGET App. The process flow corresponding to the flow numbers in Fig. 1 is as follows.

1. Acquire the standard time by the NTP server and correct the system time.

2. Send an initial query from the CENTER App to each target computer sequentially.
3. Acquire standard time by the TARGET App of each target computer that has received an initial query from the CENTER App.
4. Correct own system time by the TARGET App to the latest standard time.
5. After a specified interval, correct own system time by the NTP server from the standard time source.
6. Send a regular query sequentially from the CENTER App to each target computer.
7. Acquire standard time from NTP server by TARGET App of each target computer that received a regular query from the CENTER App.
8. Acquire the differential data by the TARGET App between the system time and the standard time.
9. Acquire CPU core temperature data by the TARGET App.
10. Repeat steps 5–9 a specified number of times.
11. Analyze the accumulated data and derive correlations to determine the characteristics.

In this case, the standard time source referenced by the criterion computer should be accurate so as to increase the accuracy of the feature data. Examples of source are the Stratum 1 NTP server or the NTP server which receives the Global Navigation Satellite System (GNSS) signal that provides the reference clock via atomic clock. The target computers are placed in the same network segment with low latency. The time drift is defined as the difference between the system time managed by the OS of the target computer and the standard time provided by the criterion computer as an NTP server. Since this time drift varies with temperature, the criterion computer collects the temperature value along with the time drift value.

The method for deriving the correlation from the acquired data (step 11) is as follows. First, let $D(t)$ be the time drift that occurs from time 0 to t, and $T(t)$ be the CPU core temperature at time t. Since the time drift D takes a continuous value, this derivative can be written as

$$\frac{dD(t)}{dt} = d(t). \tag{1}$$

Assuming that the time drift is linearly affected by the core temperature, this drift can be written as a function of temperature T, as

$$d(T) = aT + b, \tag{2}$$

where a and b are constants. Since the temperature T is a function of time t, $T(t)$, it is expressed as

$$d(t) = aT(t) + b. \tag{3}$$

However, since the values obtained in the experiments are discrete, we use the time drift per unit of time, which represents the measurement interval, and the CPU core temperature. When the i-th measurement takes place at time t_i, the average $\overline{T_i}$ of the time drift ΔD_i and the core temperature per unit time during the i, $i+1(1 \leqq i)$-th measurement is expressed as

$$\Delta D_i = D(t_{i+1}) - D(t_i), \tag{4}$$

$$\overline{T_i} = \frac{T(t_{i+1}) + T(t_i)}{2}. \tag{5}$$

Using these, we replace Eq. (3) with

$$\Delta D_i = a\overline{T_i} + b. \tag{6}$$

For N feature data consisting of a pair of drift and average CPU core temperature per unit of time obtained from the experiment, we find the slope a and the intercept b of the regression line representing the pair of ΔD and $\overline{T_i}$ at some period $P(2 \leq P \leq N)$. For the regression line

$$D = aT_i + b, \tag{7}$$

parameters a and b are obtained by the least-squares method for the corresponding number of data. The combination of averages a_{std} and b_{std} for each of these parameters a and b is the reference feature of the computer.

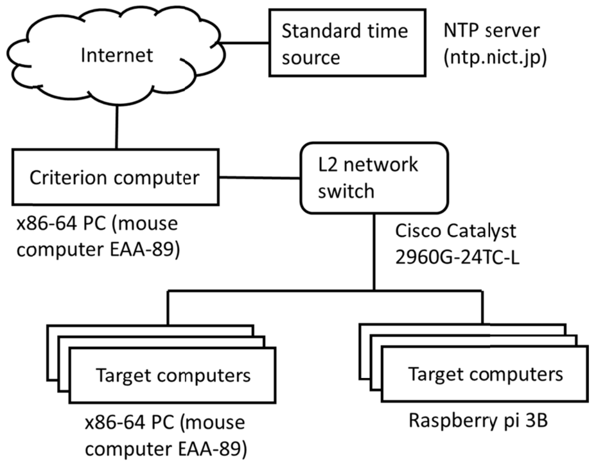


FIGURE 2. Overview of experimental system.

B. EXPERIMENTAL SYSTEM CONFIGURATION

An overview of the experimental system is shown in Fig. 2. The criterion computer consists of a PC installed with Ubuntu Server 16.04.4 and runs the NTP server and CENTER App. The NTP server is a service using the NTP daemon, which is standard in Linux, and corrects the standard time by referring to the Stratum1 NTP server provided by the National Institute of Information and Communications Technology (NICT). The CENTER App logs in to the target computer as a secure shell (SSH) client at 10-minute intervals set by the *crontab* service. Then, the shell script is executed to measure the time drift and CPU core temperature data. In this experiment, we use two types of target computer:

1) x86-64 PC, Intel Celeron T1600, Ubuntu Server 16.04.4, and

2) Raspberry pi 3B, Broadcom BCM2837 1.2 GHz 64-bit quad-core ARMv8 Cortex-A53, Raspbian (April 2018, Kernel version 4.14).

The TARGET App is installed in each of these target computers. It consists of an SSH server that accepts a query from the criterion computer and a shell script that executes the data acquisition process. In our experiments, three computers are set up for each of the two types, and all of them have the same software settings. We label these computers PC A, B, C and Raspi A, B, C. Since these target computers do not execute any processes other than the TARGET App, the load on the CPU is almost negligible. We also installed a Raspberry pi 3 equipped with a temperature sensor to check the correlation between the CPU core temperature and room temperature. It incorporates a digital temperature sensor (Maxim Integrated Products DS18B20) and records the temperature of the room where the target computers are installed.

III. RESULTS

First, we show the results of the time drift per unit of time in the experimental configuration described in Methods, without temperature information, which is a conventional method. Next, the results in which temperature was added to the above results are presented. Finally, these data are used to derive the parameters of the regression line.

A. TIME DRIFT WITHOUT TEMPERATURE INFORMATION

Figure 3 shows the time drift ΔD every ten minutes for about 50 days. This graph roughly indicates that ΔD was different for five of the six computers. However, Raspi A and Raspi C were very close, and in this case it was difficult to distinguish them using the methods that measure only time drift [8]. In contrast, none of the computers had an ΔD that was constant with respect to elapsed time, and waves could therefore be observed. However, since the rise and fall of the waves were linked to all devices, we can assume they were dependent on common environmental changes such as temperature, rather than on device-specific causes.

B. TIME DRIFT WITH CPU CORE TEMPERATURE INFORMATION

Figure 4 shows a graph obtained by excluding the outliers from the data in Fig. 3 by means of a test based on the quartiles and by using Eqs. (4) and (5), where the vertical axis is the drift ΔD per unit time and the horizontal axis is the CPU core temperature $\overline{T_i}$. In this graph, we can see different trends in Raspi A and Raspi C, which were difficult to distinguish in Fig. 3.

C. DERIVATION OF REGRESSION LINE PARAMETERS

The regression lines and correlation coefficients derived from the data in Fig. 4 are listed in Table 1. The value of the correlation coefficient r for all computers was $|r| \geqq 0.7$ and the regression line showed a strong positive correlation. Slope a and intercept b of this regression line indicate the characteristic features specific to each computer. We can see here that they had distinctly different parameters despite being the same computer type.

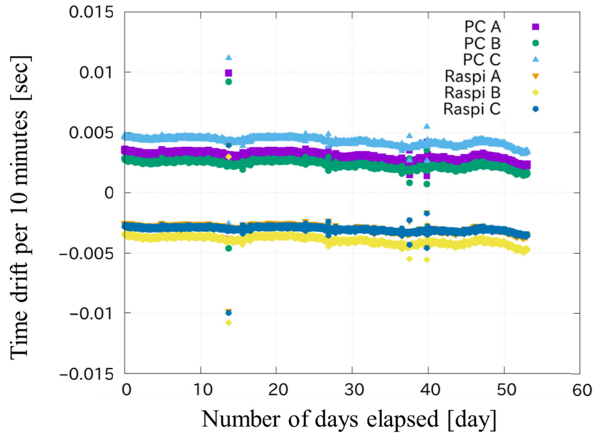


FIGURE 3. Change in the amount of time drift every ten minutes.

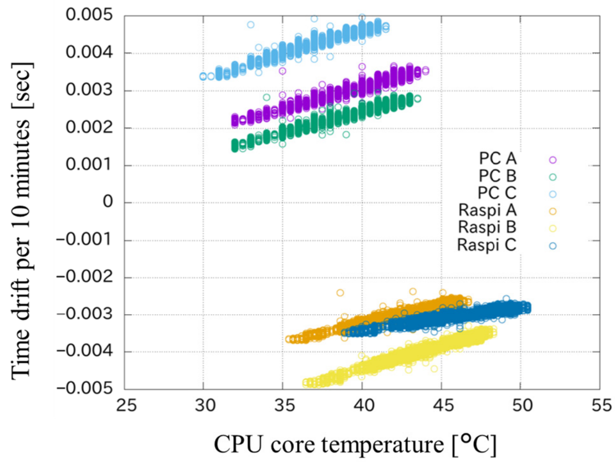


FIGURE 4. Correlation between time drift and CPU core temperature.

TABLE 1. Regression line parameters.

Computer Labels	Slope a	Intercept b	Correlation coefficient r
PC A	**1.1801902×10^{-4}**	**−0.0015995**	0.97862
PC B	**1.1429236×10^{-4}**	**−0.00210567**	0.97876
PC C	**1.2357967×10^{-4}**	**−0.0003075**	0.97640
Raspi A	**9.591916×10^{-5}**	**−0.0070656**	0.97277
Raspi B	**1.2054141×10^{-4}**	**−0.0092577**	0.98154
Raspi C	**6.111466×10^{-5}**	**−0.0058447**	0.93941

IV. DISCUSSION

A. DISCUSSION OF EXPERIMENTAL RESULTS

In this research, we use the CPU core temperature to derive the characteristic equations. First, we investigate whether it can be used in place of a temperature sensor. In general, when the computational load of a CPU increases, the power consumption of the CPU unit increases, so the CPU core temperature also increases. In this experimental configuration, data processing other than feature acquisition is not performed. Therefore, we can assume that the change in CPU core temperature depends only on the external environment,

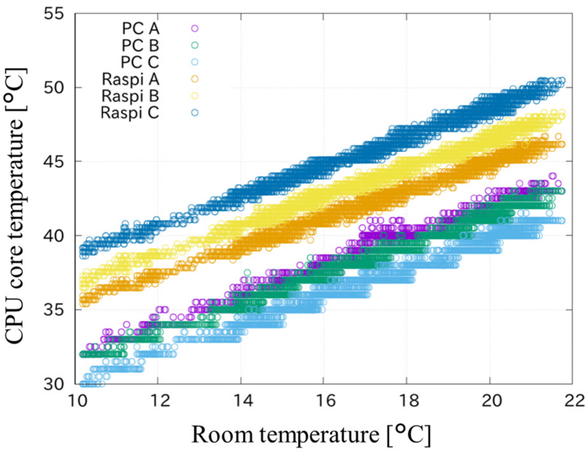


FIGURE 5. Correlation between CPU core temperature and room temperature.

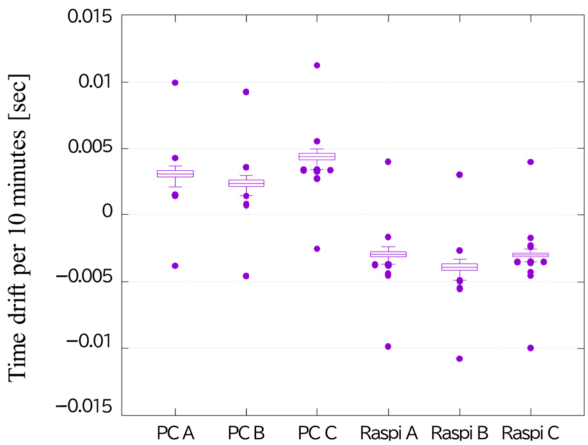


FIGURE 6. Boxplot analyzing anomalies by quartiles.

i.e., room temperature. In order to verify this, we examine the correlation between each CPU core temperature and room temperature, as shown in Fig. 5. We can see that the CPU core temperature was proportional to room temperature for all devices. Specifically, it was about 20 Celsius degree (°C) for Raspberry pi and 25 to 30°C for PC, which is higher than room temperature. These results demonstrate that the CPU core temperature can be transformed into room temperature at low load status.

Next, we discuss the causes of the outlier in the experimental data. Table 2 lists the results of quartile analysis of the experimental data, and Fig. 6 shows a boxplot based on this analysis. The interquartile range of the time drift per measurement for each computer was less than 0.5 milliseconds, but some significant outliers of about 1 to 7 milliseconds were recorded against the median. When we compare these outliers to the results in Fig. 3, the time of the outlier was almost the same for all computers. Since our method sequentially acquires data from the criterion computer to each measurement target computer, if a data processing delay or network delay of the criterion computer occurs

TABLE 2. Results of evaluation of outlier values by quartiles.

Computer label	First quartile	Central value	Third quartile
PC A	0.002821	0.003064	0.003319
PC B	0.002127	0.002339	0.002604
PC C	0.004093	0.004335	0.004582
Raspi A	–0.003136	–0.0029565	–0.002764
Raspi B	–0.004146	–0.0039225	–0.003654
Raspi C	–0.003131	–0.003003	–0.002873

during the execution, all the observations will be affected simultaneously. Therefore, these anomalous values were likely caused by errors in the external environment, which are different from the inherent characteristics of the target computer.

B. CONTRIBUTION OF THIS RESEARCH

Next, we discuss the contribution of this research in comparison to previous studies. Several prior studies have examined the relationship between clock frequency signals and temperature in electronic devices. Crystal oscillators are generally used as clock oscillators for electronic components such as CPUs and hardware clocks, and it has long been known that the frequency response of such crystal transmitters varies with temperature depending on the method used to cut the crystal [9]–[12]. The correlation of frequency with temperature has also been studied [13], and a Temperature Compensated Crystal Oscillator (TCXO) and compensation methods using temperature characteristics have been proposed [14], [15]. However, these techniques are used to generate a stable clock frequency signal and cannot be used to obtain the characteristics of the computer. Other studies have shown the relationship between temperature, PC CPU clock frequency, and clock drift [16], [17], and Marouani *et al.* reported that the clock frequency varied by approximately 150 hertz per Celsius degree for the PC CPUs in their study [18]. However, they did not mention any individual clock characteristics of the computer. The key contribution of our research is the experimental results showing that the time drift caused by the clock signal and its correlation with temperature have different characteristics for individual computers. By approximating this to a regression line, we developed a method to treat these parameter values as characteristic features of each computer. As our method utilizes the CPU core temperature, which is easily obtained in many digital devices such as PCs and single-board computers for IoT, it does not require the implementation of a special temperature sensor. As such, it can be used with existing devices simply by introducing the relevant software, and thus has good potential for contributing to the spread of IoT infrastructure in the future.

C. FUTURE CHALLENGES

Finally, we briefly touch on the future challenges of this research. The proposed method was derived by linearly approximating the correlation equation, but the temperature characteristics of quartz crystal units show nonlinear characteristics in the low and high temperature ranges [19]. In future work, it will be necessary to analyze the practical range of applicability of the linear approximation and/or to determine how to derive correlations in the nonlinear domain. In addition, the results of this experiment were conducted under no-load conditions to suppress any change in the CPU core temperature. However, in practical operation, various data processing programs are running and the computational load on the computer is heavy. Since this increased load induces an increase in the temperature of the CPU core, we need to resolve this issue for practical use. From another practical point of view, it is necessary to consider the aging of electronic devices. Although the results in this study are based on the data of about 50 days, the characteristics of quartz crystal units also change [20], so we need to investigate the effect of aging on the characteristic values in order to accurately determine the effect of long-term operation.

V. CONCLUSION

Proper management of various types of computers is essential for the secure use of IoT services, and versatile and simple identification techniques will be significant to achieving this. The contribution of this research is our development of a method that extracts new hardware-inherent characteristic information for use in identifying computers. Our method samples the temperature and system time deviation of the computer and then derives the inherent characteristic. Using this method, we obtained several unique characteristic expressions for each of six computers with the same software configuration. A key feature of the proposed method is its ability to use CPU core temperature instead of temperature sensors along with a simple regression equation to describe the intrinsic characteristics of the computer. This means that it does not require the installation of any special hardware on the computer and can be used by implementing software that does simple computational work. This makes it a highly versatile method that can be implemented on computers already installed in the field. Moreover, it has the potential to be used in a wide range of future IoT platforms. For its practical use, it is necessary to create a system that can identify the different types of computers utilized in a general network environment, rather than just the current one used in our experiment. In future work, we plan to improve the accuracy of the identification method, expand the number of computer variations, and enable long-term operation, thereby achieving practical computer identification technology.

REFERENCES

[1] D. Evans, "The Internet of Things. How the next evolution of the Internet is changing everything," in *Proc. Cisco IBSG*, Apr. 2011, pp. 1–11. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[2] (2017). *Number of Connected IoT Devices Will Surge to 125 Billion by 2030*. HIS Markit News Release. Accessed: Sep. 23, 2020. [Online]. Available: http://news.ihsmarkit.com/prviewer/release_only/slug/number-connected-iot-devices-will-surge-125-billion-2030-ihs-markit-says

[3] I. Garcia-Magarino, S. Sendra, R. Lacuesta, and J. Lloret, "Security in vehicles with IoT by prioritization rules, vehicle certificates, and trust management," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 5927–5934, Aug. 2019, doi: 10.1109/JIOT.2018.2871255.

[4] I. García-Magariño, G. Palacios-Navarro, R. Lacuesta, and J. Lloret, "ABSCEV: An agent-based simulation framework about smart transportation for reducing waiting times in charging electric vehicles," *Comput. Netw.*, vol. 138, pp. 119–135, Jun. 2018, doi: 10.1016/j.comnet.2018.03.014.

[5] O. Bello and S. Zeadally, "Toward efficient smartification of the Internet of Things (IoT) services," *Future Gener. Comput. Syst.*, vol. 92, pp. 663–673, Mar. 2019, doi: 10.1016/j.future.2017.09.083.

[6] B. Wang, M. Li, X. Jin, and C. Guo, "A reliable IoT edge computing trust management mechanism for smart cities," *IEEE Access*, vol. 8, pp. 46373–46399, 2020, doi: 10.1109/ACCESS.2020.2979022.

[7] (2019). *W3C and FIDO Alliance Finalize Web Standard for Secure, Passwordless Logins*. [Online]. Available: https://www.w3.org/2019/03/pressrelease-webauthn-rec.html.en

[8] N. Hoshikawa, T. Shimobaba, and T. Ito, "Trait extraction method for digital equipment using clock drifts based on comparison of criterion server," in *Proc. 2nd Int. Conf. Comput. Commun. Internet (ICCCI)*, Jun. 2020, pp. 190–194, doi: 10.1109/ICCCI49374.2020.9145998.

[9] R. D. Mindlin, "Thickness-shear and flexural vibrations of crystal plates," *J. Appl. Phys.*, vol. 22, no. 3, pp. 316–323, Mar. 1951, doi: 10.1063/1.1699948.

[10] R. D. Mindlin, "High frequency vibrations of crystal plates," *Quart. Appl. Math.*, vol. 19, no. 1, pp. 51–61, Apr. 1961.

[11] Y.-K. Yong and J. T. Stewart, "Mass-frequency influence surface, mode shapes, and frequency spectrum of a rectangular AT-cut quartz plate," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 38, no. 1, pp. 67–73, Jan. 1991, doi: 10.1109/58.67837.

[12] Y. Jing, J. Chen, X. Chen, and X. Gong, "Frequency shift of thickness-shear vibrations of AT-cut quartz resonators due to a liquid layer with the electrode stiffness considered," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 54, no. 7, pp. 1290–1292, Jul. 2007, doi: 10.1109/TUFFC.2007.389.

[13] R. J. Holbeche and P. E. Morley, "Investigation into temperature variation of equivalent-circuit parameters of AT-cut quartz crystal resonators," *IEE Proc. A, Phys. Sci., Meas. Instrum., Manage. Edu. Rev.*, vol. 128, no. 7, pp. 507–510, Oct. 1981, doi: 10.1049/ip-a-1.1981.0077.

[14] V. Stofanik, I. Balaz, and M. Minarik, "An improvement in the temperature-compensated DDS," in *Proc. IEEE Int. Freq. Control Symp. PDA Exhib.*, May 2002, pp. 662–664, doi: 10.1109/FREQ.2002.1075964.

[15] H.-W. Peng, C.-H. Su, P. C.-P. Chao, J.-W. Hsieh, and C.-K. Chang, "A new small-sized pierce crystal oscillator readout with novel on-chip all-digital temperature sensing and compensation," in *Proc. IEEE Sensors*, Nov. 2014, pp. 225–228, doi: 10.1109/ICSENS.2014.6984974.

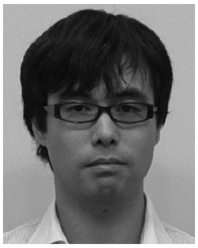
[16] R. K. Karlquist, L. S. Cutler, E. M. Ingman, J. L. Johnson, and T. Parisek, "A low-profile high-performance crystal oscillator for timekeeping applications," in *Proc. Int. Freq. Control Symp.*, May 1997, pp. 873–884, doi: 10.1109/FREQ.1997.639205.

[17] H. Kawashima and K. Sunaga, "Temperature compensated crystal oscillator employing new shape GT cut quartz crystal resonator," in *Proc. 45th Annu. Symp. Freq. Control*, May 1991, pp. 410–417, doi: 10.1109/FREQ.1991.145929.

[18] H. Marouani and M. R. Dagenais, "Internal clock drift estimation in computer clusters," *J. Comput. Syst., Netw., Commun.*, vol. 2008, pp. 1–7, May 2008, doi: 10.1155/2008/583162.

[19] L. C. Barcus, "Nonlinear effects in the AT-cut quartz resonator," *IEEE Trans. Sonics Ultrason.*, vol. 22, no. 4, pp. 245–250, Jul. 1975, doi: 10.1109/T-SU.1975.30803.

[20] S. Yamagata, "Aging characteristic of light-modulation quartz-crystal oscillator with glass-bar-mounted plano-convex AT-cut quartz-crystal plate," *Electron. Lett.*, vol. 19, no. 19, pp. 795–797, Sep. 1983, doi: 10.1049/el:19830542.



NAOTO HOSHIKAWA received the B.E. degree in engineering from Chiba University, in 2004, and the M.E. and Ph.D. degrees in information science from Nagoya University, in 2006 and 2012, respectively. He joined NTT EAST, in 2009, where he was responsible for developing the server infrastructure for telecommunications services. He moved to NTT Network Service Systems Laboratories, in 2013, to research and develop communication service systems and service infrastructures for IoT. From 2018 to 2019, he was an Assistant Professor with the National Institute of Technology, Oyama College, Japan, where he has been a Lecturer, since 2019. His research interests include information security, the IoT, network computing, and their service applications. He is a member of the IEICE and IPSJ.

RYO NAMIKI received an Associate degree from the National Institute of Technology, Oyama College, in 2019. He is currently pursuing the B.E. degree in computer science with the Tokyo University of Agriculture and Technology. His research interests include information systems and security.

KATSUMI HIRATA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in systems and information engineering from the University of Tsukuba, Japan, in 1997, 1999, and 2002, respectively. From 2002 to 2015, he was an Assistant Professor with the National Institute of Technology, Oyama College, Japan, where he has been an Associate Professor, since 2015. His research interests include acoustic environment identification, speech signal processing, random signal processing, and computer aided education. He is a member of SICE, IEICE, and ASJ.

ATSUSHI SHIRAKI received the B.E. degree in electrical and electronic engineering from Yamaguchi University, Japan, in 2003, and the M.E. and Ph.D. degrees from Chiba University, Japan, in 2005 and 2008, respectively. From 2008 to 2011, he was an Assistant Professor with the National Institute of Technology, Kisarazu College, a Lecturer, from 2011 to 2014, and an Associate Professor, from 2014 to 2015. Since 2015, he has been an Associate Professor with the Institute of Management and Information Technologies, Chiba University. His research interests include 3D systems, educational technology, and their applications. He is a member of ITE, IEICE, IPSJ, and JSET.

TOMOYOSHI ITO received the B.S. degree in pure and applied sciences and the M.S. and Ph.D. degrees in earth science and astronomy from The University of Tokyo, Japan, in 1989, 1991, and 1994, respectively. From 1992 to 1994, he was a Research Associate with Gunma University, Japan, and an Associate Professor, from 1994 to 1999. From 1999 to 2004, he was an Associate Professor with the Graduate School of Engineering, Chiba University, Japan, where he has been a Professor, since 2004. His research interests include high-performance computing and its applications, such as electronic holography for 3D TV. He is a member of ACM, ASJ, ITE, IEICE, IPSJ, and OSA.