ISSN 1345-3041

Dec.2025 / Vol.192

Mitsubishi Electric

ADVANCE

Digital Technologies for Sustainable Business Growth (The first part)

MITSUBISHI ELECTRIC
Changes for the Better

December 2025 / Vol. 192
Mitsubishi Electric ADVANCE

Digital Technologies for Sustainable Business Growth (The first part)

• Editorial Inquiries
Fumitoshi Yoshikawa
Corporate Productivity Planning & Engineering Dept.
Fax: +81-3-3218-2465

Mitsubishi Electric Advance is published on line quarterly (in March, June, September, and December) by Mitsubishi Electric Corporation.

CONTENTS

Overview

Digital Technologies for Sustainable Business Growth 1
by *Toru Oka*

Technical Reports

A Cutting-Edge Digital Gate Drive Technology for High-Efficiency Power Semiconductor Operation 2
by *Kenichi Morokuma, Makoto Takamiya*

System Control Planning Technology that Supports Automated Production Site Operations 7
by *Atsuko Nakai, Takanobu Yaguchi*

A Formal Verification Using Graph-Database for Security Protocols 14
by *Hisashi Mori, Kazuki Yonemochi, Manabu Misawa*

Insider Threat Detection Using Decoy 19
by *Takumi Yamamoto, Kohei Nozawa*

Opto-electronic Convergence Technology for Achieving High Capacity and Low Power Consumption in Next-generation Data Centers 24
by *Nobuo Ohata, Mizuki Shirao*

Precis

The Mitsubishi Electric Group positions sustainability as a cornerstone of its management and is transforming itself into an "Innovative Company" that creates new value without fear of taking on risks. The evolution of digital technologies is the key to transforming into an Innovative Company. We will introduce specific initiatives related to advanced digital technologies in two parts: Part 1 (this issue, Mitsubishi Electric ADVANCE Vol. 192, December 2025) and Part 2 (next issue, Vol. 193, March 2026).

Overview

Author: *Toru Oka**

** Executive Officer, In charge of Intellectual Property, Vice President, Corporate Research and Development*

Digital Technologies for Sustainable Business Growth

The Mitsubishi Electric Group places sustainability at the core of its management and is transforming itself into an “Innovative Company” that creates new values without fear of taking risks. We will leverage component and digital technologies to advance R&D for sustainable business growth. We will also focus on developing Foresight Technology that addresses social challenges and creates new values.

Digital technologies are the key to transformation into an Innovative Company. By leveraging “Serendie,” we will improve the efficiency and safety of products and systems based on data, enabling the creation of new business models. We will also develop “Intelligent autonomous control technologies” for integrated operation of complex systems and strengthen “Cybersecurity technologies” that minimize the impact of attacks and enhance defensive capabilities. Additionally, we will focus on “AI and Generative AI technologies” that leverage physical models and simulations to achieve fast, high-accuracy inference and automation. Going forward, we will concentrate on “Neuro-Physical AI,” “Secure AI,” and “Agent AI” to provide new value in a wide range of business domains. In addition, we will invest in the development of “Foresight Technology”—technology that will have a major impact on society and business—to drive future business growth. In the December and March issues of Mitsubishi Electric ADVANCE, we will present specific initiatives related to these digital technologies.

We will continue to promote innovative R&D, address increasingly diverse and serious social challenges, and contribute to the realization of a sustainable society.

A Cutting-Edge Digital Gate Drive Technology for High-Efficiency Power Semiconductor Operation

Authors: *Kenichi Morokuma*, Makoto Takamiya***

**Advanced Technology R&D Center, ** University of Tokyo*

Abstract

In pursuit of carbon neutrality by 2050, the presence of power electronics equipment is growing, driven by the expansion of renewable energy and other factors. In power electronics equipment, the power loss (switching energy loss) and electromagnetic noise generated by the switching operation of power devices have a trade-off relationship. Digital gate drive technology is drawing attention as a power-device driving technique to improve this trade-off relationship.

In this work, through joint research with the University of Tokyo, the authors proposed a drive method that changes the signal strength for driving power devices at optimal timing using a general-purpose gate driver integrated circuit (IC). Compared with conventional drive methods, the proposed drive method achieved a reduction in turn-on switching energy loss of 25% and 18% at load currents of 50 A and 100 A, respectively. The application of the proposed digital gate drive technology is expected to contribute to further energy savings in power electronics equipment.

1. Introduction

Digital gate drive technology that changes the drive signal strength in multiple steps during the switching operation of power devices is drawing attention. This technology makes it possible to improve the trade-off relationship between switching loss and electromagnetic noise in power modules. By optimizing the timing for varying the drive signal strength in accordance with operating conditions such as load current I_L and temperature, the technology contributes to energy savings in power electronics equipment[(1)]. Figure 1 shows a method for determining the timings t_1 and t_2 at which to change the drive signal strength[(1) (2) (3)] with respect to the gate-emitter voltage V_{GE} and collector current Ic during turn-on operation of the power module. In this study, the authors propose a novel timing determination method that changes the drive signal strength to “strong, weak (high impedance), and strong” and determines the timings t_1 and t_2 at which to change the drive signal strength. Furthermore, the authors verified the validity of the proposed method based on the evaluation results using a general-purpose gate driver integrated circuit (IC).

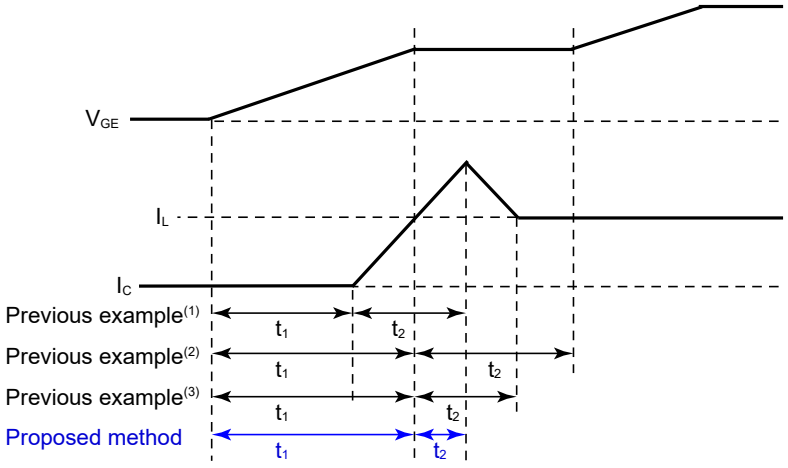
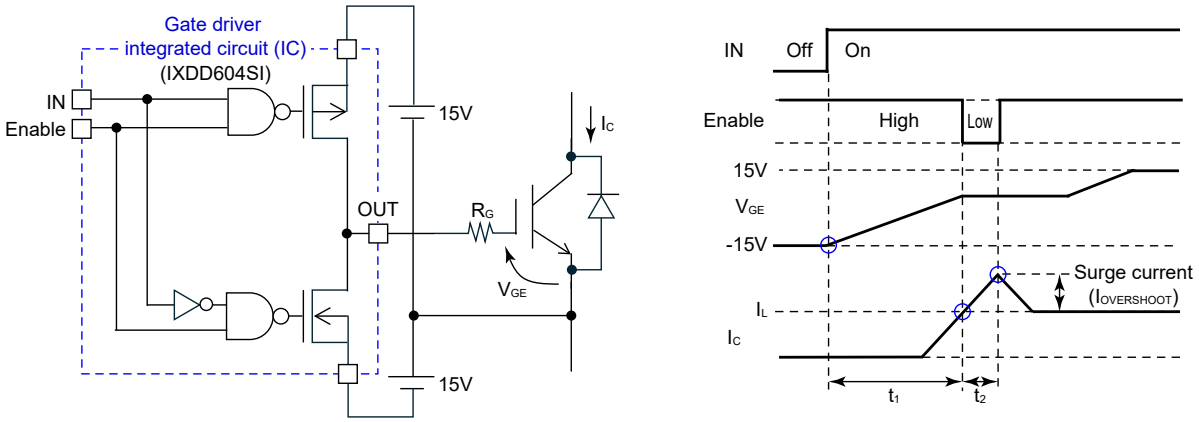


Fig. 1 Various methods for determining t_1 and t_2 at turn-on

2. Proposed Method for Determining Optimal Drive Timing

Figure 2 shows the circuit diagram and timing chart of the proposed method. The proposed method uses a gate driver integrated circuit (IC) which has Enable function (IXDD604SI). The timing t_1 and t_2 in the timing chart are controlled by the input signal IN and the Enable signal. When the Enable signal is low, the output OUT becomes high impedance. In the proposed method, t_1 is the period from the rising edge of V_{GE} to the timing when I_C reaches the load current I_L, and t_2 is the period from the timing at which I_C reaches the load current I_L to the timing at which I_C reaches its peak value. The values of t_1 and t_2 change as the operating conditions of the power module and they are calculated from the measured V_{GE} and I_C waveforms.



(a) Circuit diagram (b) Timing chart

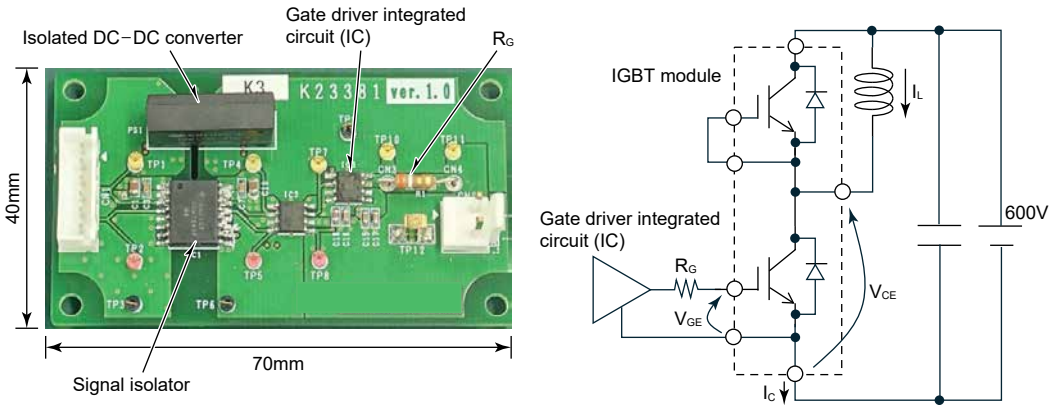
Fig. 2 Circuit diagram and timing chart of the proposed method

The loss reduction rate in the proposed method is defined by Equation (1), where $E_{LOSS,\ DGD}$ denotes the switching energy loss in the proposed method, and $E_{LOSS,\ CONV}$ denotes the switching energy loss in the conventional gate drive circuits (hereafter referred to as the “conventional method”). $E_{LOSS,\ DGD}$ is obtained with the measurement results. $E_{LOSS,\ CONV}$, on the other hand, is calculated using the trade-off curve, that is measured by varying the gate resistance R_G with conventional method, between energy loss E_{LOSS} and surge current $I_{OVERSHOOT}$.

$$\text{Loss reduction rate} = \frac{E_{LOSS,\ CONV} - E_{LOSS,\ DGD}}{E_{LOSS,\ CONV}} \times 100 \quad \cdots\cdots (1)$$

3. Evaluation Method

Figure 3 shows the gate driver board and the circuit diagram for switching evaluation. The gate driver board is equipped with a signal isolator and an isolated DC-DC converter. An Insulated Gate Bipolar Transistor (IGBT) module (CM100DY-24T, rating: 1,200 V, 100 A) was used as a device under test. In this study, t_1 and t_2 were measured using the determination methods of the proposed method and the conventional method, as well as the previous examples[1][2][3] respectively. In the proposed method, t_1 and t_2 were measured with 2,400 combinations at 2 ns intervals.

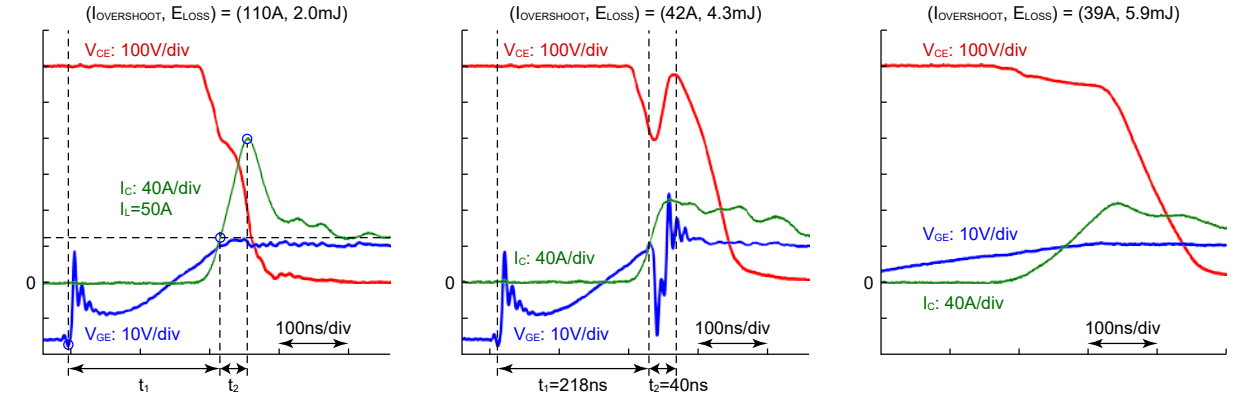


(a) Gate driver board (b) Switching evaluation circuit diagram

Fig. 3 Gate driver board and switching evaluation circuit diagram

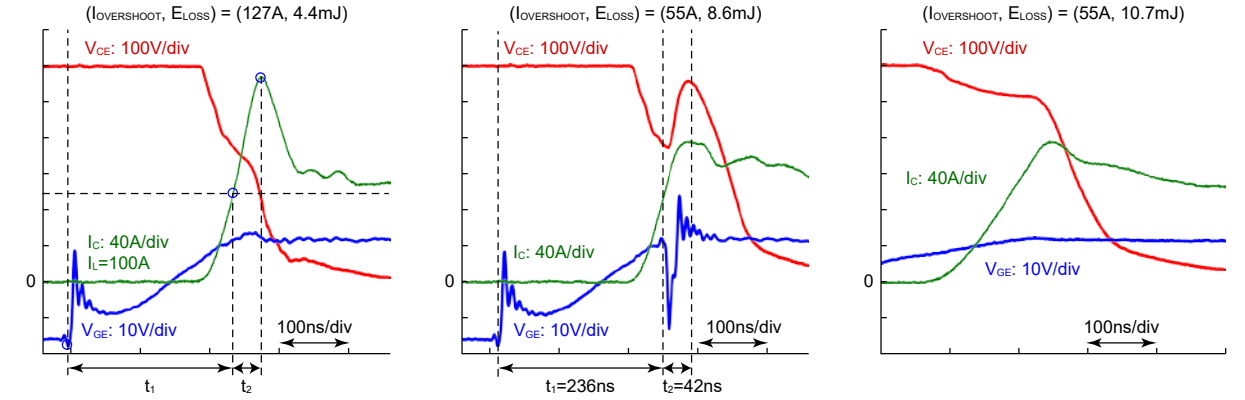
4. Evaluation Results

Figure 4 and Fig. 5 show the switching waveforms for the proposed method and the conventional method at load currents I_L =50 A and 100 A, respectively. For the proposed method, t_1 and t_2 were calculated from the switching waveform on condition that the gate resistance R_G of the conventional method was set to 3.9 Ω. The switching energy loss and surge current were obtained from these switching waveforms, and the loss reduction rate was compared.



(a) Conventional method Point A1 (R_G=3.9Ω) (b) Proposed method (c) Conventional method Point B1 (R_G=18Ω)

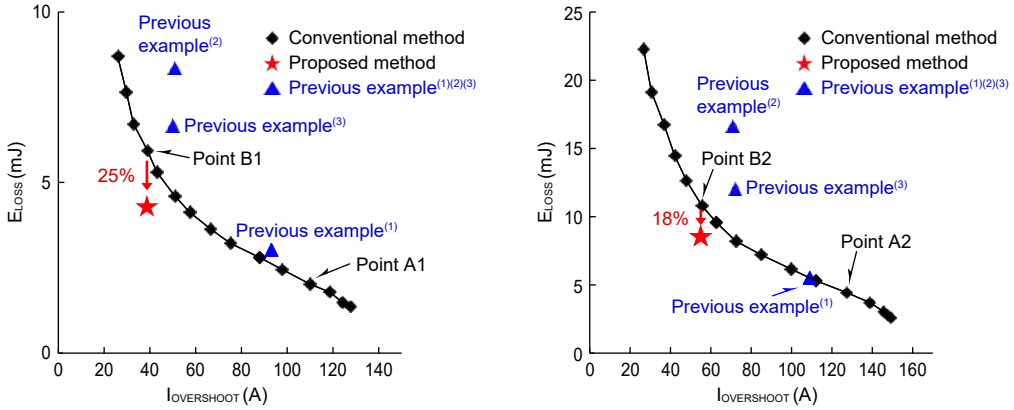
Fig. 4 Switching waveforms of the proposed method and the conventional method at I_L=50 A



(a) Conventional method Point A2 (R_G=3.9Ω) (b) Proposed method (c) Conventional method Point B2 (R_G=12Ω)

Fig. 5 Switching waveforms of the proposed method and the conventional method at I_L=100 A

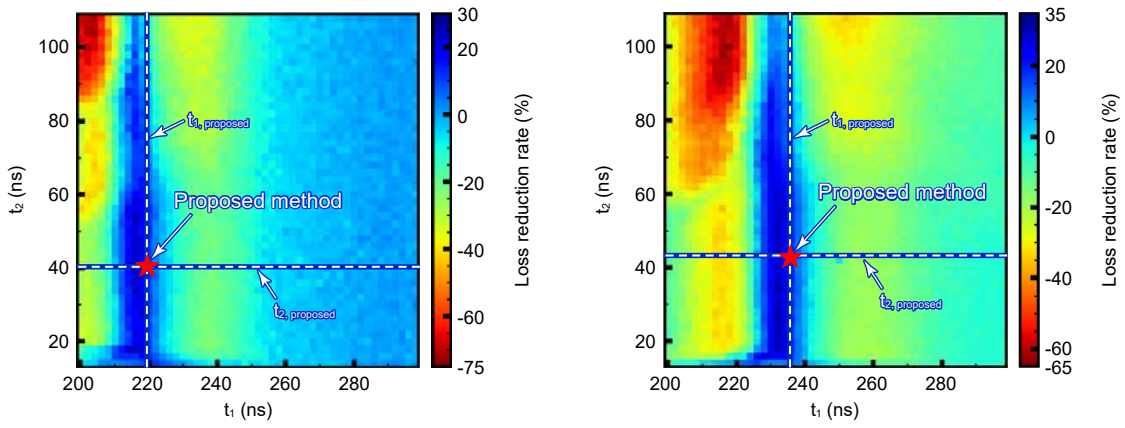
Figure 6 shows the evaluation results for the proposed method, the conventional method, and previous examples[1][2][3] of switching energy loss E_{LOSS} and surge current $I_{OVERSHOOT}$ at load currents I_L = 50 A and 100 A. With the proposed method, while maintaining the surge current $I_{OVERSHOOT}$ at about the same level as the conventional method, E_{LOSS} was reduced by 25% at load current I_L = 50 A, and 18% at I_L = 100 A, respectively. Note that E_{LOSS} in the previous examples[1][2][3] was higher than that in the conventional method when compared on condition that $I_{OVERSHOOT}$ matched that of the conventional method, indicating a deterioration in switching loss.



(a) I_L=50A (b) I_L=100A

Fig. 6 Evaluation results of switching loss E_{LOSS} and surge current $I_{OVERSHOOT}$

Figure 7 shows the dependence of the loss reduction rate on t_1 and t_2 at load current I_L = 50 A and 100 A. Optimizing t_1 is important because the loss reduction rate depends significantly on t_1 while the dependence on t_2 is small. With the proposed method, the authors were able to obtain the optimal value of t_1, which has a large impact on the loss reduction rate and achieved a higher loss reduction effect. Table 1 shows the evaluation results of the loss reduction rate at temperatures T_j of 25°C, 75°C, and 125°C for the proposed method and previous examples[1][2][3]. Even under conditions with temperatures T_j of 75°C and 125°C, the authors confirmed that the proposed method achieves a higher loss reduction effect than previous examples[1][2][3].



(a) Load current I_L=50A (b) Load current I_L=100A

Fig. 7 Dependence of loss reduction rate on t_1 and t_2

Table 1 Loss reduction rate of the proposed method and previous examples[1][2][3]

	I_L=50 A			I_L=100 A		
	T_j=25°C	T_j=75°C	T_j=125°C	T_j=25°C	T_j=75°C	T_j=125°C
Proposed method (%)	25	23	16	18	26	27
Previous example[1] (%)	-20	-10	-9	0	-2	-1
Previous example[2] (%)	-84	-64	-50	-95	-96	-70
Previous example[3] (%)	-43	-75	-64	-45	-78	-90

5. Conclusion

The authors proposed a novel method for digital gate drive technology to determine the timing for adjusting the strength of the drive signal. The proposed method achieved the maximum loss reduction effect, demonstrating its validity.

The authors will continue to develop the proposed method toward practical implementation, contributing to energy savings in power electronics equipment.

References

(1) Zhang, F., et al.: Advanced Active Gate Drive for Switching Performance Improvement and Overvoltage Protection of High-Power IGBTs, IEEE Transactions on Power Electronics, 33, No.5, 3802–3815 (2017)
(2) Camacho, P.A., et al.: A Novel Active Gate Driver for Improving SiC MOSFET Switching Trajectory, IEEE Transactions on Industrial Electronics, 64, No.11, 9032–9042 (2017)
(3) Manuel, R., et al.: Implementation of Current-Source Gate Driver with Open-Loop Slope Shaping for SiC-MOSFETs, International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, 1–8 (2021)

System Control Planning Technology that Supports Automated Production Site Operations

Authors: *Atsuko Nakai*, Takanobu Yaguchi**

**Advanced Technology Research Center*

Abstract

There is a growing need for automated operations at production sites to compensate for labor shortages. Production schedules at production sites are often created by site supervisors, but this has become an extremely labor-intensive task. To address this issue, we developed a method that, when production capacity drops at production sites where factory workers and production equipment work together, automatically devises and applies the most cost-effective countermeasures by using predefined profile information for factory workers and for production equipment. We also developed an optimization method that combines a formulation part as an integer optimization problem and a rule-based part, in order to rapidly create production schedules that minimize downtime for production equipment.

1. Introduction

In recent years, at production sites in Japan, innovations in production technology have been required, such as a shift to high-mix, low-volume production due to a declining working-age population and the diversification of consumer needs[(1)]. In particular, at sites such as food factories and logistics sites, nighttime and holiday operations are indispensable, making it even more difficult to secure labor. Automation of production equipment is advancing to compensate for labor shortages. Reducing the number of factory workers for labor-saving is expected to ensure a stable workforce and lighten workloads to improve production efficiency. However, a new issue is arisen that the performance difference between production equipment and factory workers have widened, making it difficult to maintain manufacturing line performance. Aiming to improve production efficiency, site supervisors have been creating production schedules based on intuition and experience, but this has become a significant work load at production sites. To overcome these challenges, we devised a method that automatically plans and applies optimal countermeasures when production capacity declines at production sites where factory workers and production equipment coexist. Specifically, we developed an approach that leverages predefined profile information for factory workers and for production equipment to automatically derive the most cost-effective response. Furthermore, to rapidly create production schedules that minimize downtime of production equipment, the many parameters that must be considered are split into a part that performs formulation as an integer optimization and a part that processes them in a rule-based manner and then combined, to develop an optimization method that enables creating a production schedule within tens of seconds. Among the constraints required by production sites, we perform formulation as an integer optimization for the highly variable constraints and use a solver to obtain a solution. Next, the production schedule is created by applying, as rule-based algorithmic processing, the constraints that are generic to production sites to the solution obtained by the solver.

2. Production System Operation Control Method Using Profile Information of Production Equipment and Factory Workers

This chapter describes the method that automatically creates control commands using the profile information defined for production equipment and for factory workers. The proposed method sequentially collects and analyzes the profile information of production equipment—comprising robots, automated transporters, controllers, etc.—and the profile information of factory workers, including status values, sensor readings, operation histories, and so on, and compares them with the profile information from past instances of similar work. This simulates how changing or not changing the profile information of production

equipment or factory workers would achieve the desired performance to suit the conditions at the site. It then determines the control details needed to achieve the desired performance and aims to distribute them to the factory workers and production equipment.

Profile information consists of attribute information, which is information specific to production equipment or specific to factory workers, and performance indicator information, which is information about the work capability exhibited by production equipment or factory workers within the production system. That is, profile information is defined as information indicating the overall work capability provided by factory workers and production equipment. Using robots as an example of production equipment, examples of specific profile information items include, as attribute information, not only catalog specification values but also information related to robot operation such as “installed line name.” Performance indicator information includes information calculated from actual operating data, such as “work speed” or “defect incidence rate,” computed for each workpiece or for each set of working conditions. For factory workers, examples of corresponding profile information items include, as attribute information, personnel information and information related to the duties they are engaged in. As with production equipment, performance indicator information includes information calculated from actual operating data, such as “work speed” or “defect incidence rate,” computed for each task or for each set of working conditions. While profile information varies by factory type and by differences in installed equipment, a minimum level of standardization is necessary. Therefore, as a template for profile information, OPC UA*1 information models and companion specifications will be utilized. For example, OPC UA for Robotics has been established for robots. Based on these information models, by adding and extending items related to profile information, standardization of profile information can be achieved, and the costs involved in collecting and evaluating profile information can be reduced.

This method is designed to operate with an edge platform terminal installed in the factory as its core. Figure 1 shows the system configuration. Edgecross, an open software platform in the edge computing domain provided by the Edgecross Consortium, a general incorporated association(2) *2 was adopted within the edge platform terminal, to generically perform data collection, processing, and distribution on the production floor. In this case, we designed a mechanism to share profile information between OPC UA–compliant production equipment and other systems and Edgecross, and to mutually distribute it using the OPC UA API (Application Programming Interface)(3).

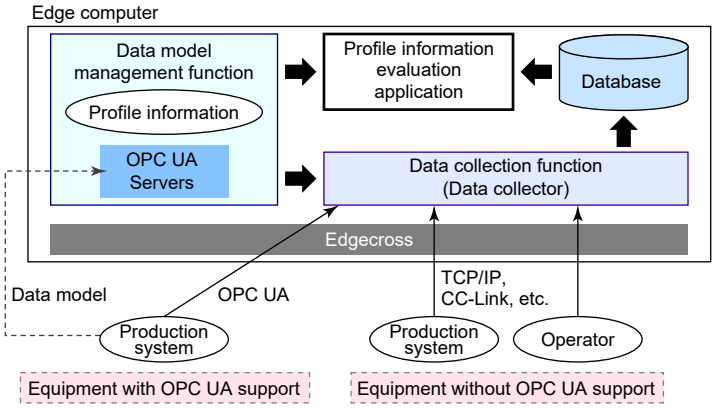


Fig. 1 System configuration

In this work, we created a profile information evaluation application within the edge platform terminal. This application sequentially evaluates the profile information collected by Edgecross and, when a drop in production system performance is anticipated, simulates the effect of issuing control commands to change the profile information items with a high improvement effect among the evaluation indicators preset by the administrator. Based on virtually generated profile information values for workers and production equipment, we ran a prototype of an application that performs sequential analysis to verify whether performance

*1 An international standard in the industrial automation field for enabling equipment and systems from different vendors to exchange data safely and reliably. OPC is a registered trademark of the OPC Foundation.

*2 Edgecross is a registered trademark of the Edgecross Consortium.

drops can be predicted and whether control commands for performance improvement can be created, and conducted verification on a commercially available edge computer. Figure 2 shows an example of the prototype verification screen.

Fig. 2 Example of verification screen for the production system operation control method

In this example, a production line in which two picking robots and multiple workers pack lunch box ingredients is simulated, with Edgecross collecting the remaining quantities of ingredients and the operating status of the robots. We verified whether control commands could be prepared in advance so that the line would not stop when the remaining quantity reaches zero while predicting the remaining quantities of ingredients. As a result, we confirmed that the evaluation of profile information and the creation of control commands could be performed with almost no delay[4].

3. Parallel Machine Scheduling Method with Processing Priorities

In this chapter, we propose a method for rapidly creating production schedules for multiple pieces of production equipment at production sites engaged in high-mix, low-volume manufacturing.

We describe the assumptions of the scheduling problem under consideration. At a production site where m machines are operating, n operations called 'jobs' are waiting to be executed. Each job j may be executed in any order, and may be processed on any production equipment. For each job j, the processing time l_j is constant regardless of which machine is used. Furthermore, we define that the maximum processing time is at most 23 hours. Each job j has a due date d_j set as the date by which completion of the job is desired. Each job j is constructed as a unit that can be processed on a single piece of production equipment. Two types of jobs are available—in addition to "regular jobs" created in advance, there are "interrupt jobs" created urgently in cases such as processing failures or insufficient processing, and each job falls into one of these. Based on these assumptions, we establish the following constraints.

Constraint.1 Minimization of total tardiness: For each job j, the due date d_j is specified on a daily basis. The difference between the due date d_j and the processing date after scheduling is defined as "tardiness"; if the processing date is before the due date, the tardiness is 0. A schedule that minimizes the total tardiness across all jobs is created.

Constraint.2 Priority constraint: Each interrupt job is assigned a priority p, and higher priority jobs are scheduled earlier time. Regular jobs are scheduled after the date and time when all interrupt jobs have completed processing.

Constraint.3 Operating time constraint: For each production equipment i, an upper limit on operating time t_i can be set. In that case, for each production equipment i, the total processing time of the jobs it processes does not exceed t_i.

Constraint.4 Load balancing constraint: A schedule that considers load balancing is created, so that, for each production equipment i, the total processing time of the jobs processed is approximately uniform.

Constraint.5 Sequential execution constraint: On each production equipment i, jobs are executed sequentially.

Here, under the assumptions stated earlier, as a method to obtain a schedule that satisfies the constraints, we adopt the policy of dividing it into a part handled by integer optimization formulation and a part processed by a rule-based algorithm.

The integer optimization processing part formulates the assumptions and the constraints from Constraint.1-4 as an integer optimization. Because Constraint.2-3 are conditions that any desired schedule must always satisfy, they are incorporated as hard constraints into the constraint equations of the integer optimization. Because Constraints.1,4 are desirable conditions to obtain a better solution, they are incorporated as soft constraints into the objective function of the optimization problem. The formulation is as follows.

$$x_{ij} = \begin{cases} 1 \text{ (assign job } j \text{ to production equipment } i) \\ 0 \text{ (otherwise)} \end{cases} \quad (1)$$

$l_j \in \mathbb{Z}$: Processing time for job j on the production equipment (2)

J : Collection of all jobs (3)

$d_j \in \{0,\ 1,\ 2,\ \cdots\}$: Number of days until due date of job j (4)

t_i : Maximum operating time of production equipment i (5)

$p \in \{1,\ 2,\ \cdots\}$: Priority assigned to each job j (6)

$J_p \subseteq J$: Collection jobs with priority p (7)

$b_p \in \{0,\ 1\}$: Auxiliary variable for priority constraints (8)

$e_j = f(d_j)$: Benefit from on-time delivery (9)

minimize $-P_1 - P_2$ (10) Objective function for Constraints.1-2

subject to

$P_1 = \Sigma_i \Sigma_j x_{ij} e_j$ (11) Constraint.1 Minimize total tardiness

$P_2 = y$ (12) Constraint.4 Load balancing constraint

$y \leq \Sigma_{j \in J} l_j x_{ij} \ (i = 1,\ 2,\ \cdots,\ m)$ (13) Constraint.4 Load balancing constraint

$\Sigma_{j \in J} l_j x_{ij} \leq t_i \ (i = 1,\ 2,\ \cdots,\ m)$ (14) Constraint.3 Operating time constraint

$\Sigma_i x_{ij} \leq 1 \ (j = 1,\ 2,\ \cdots,\ n)$ (15)

$b_p \geq b_{p+1} \ (p = 1,\ 2,\ \cdots)$ (16) Constraint.2 Priority constraint

$\Sigma_i x_{ij} \leq b_p \ (j \in J_p, p = 1,\ 2,\ \cdots)$ (17) Constraint.2 Priority constraint

$b_{p+1} \leq \Sigma_i x_{ij} \ (j \in J_p, p = 1,\ 2,\ \cdots)$ (18) Constraint.2 Priority constraint

Next, this outlines the rule-based processing part. Among the constraints, Constraint.5 is handled by a rule-based algorithm. Specifically, by solving the integer optimization expressed by formulas (1)–(18), we obtain the relationship between each job and the production equipment that processes it, and using this, we create a time-sequenced schedule for each production equipment with a rule-based algorithm. The algorithm used is shown below.

Step 1: Gather the jobs assigned to each production equipment.

Step 2: Among the gathered jobs, sort the interrupt jobs in order of higher priority. If priorities are the same, order them starting from the jobs with the shortest margin for the due date.

Step 3: After all interrupt jobs, arrange the regular jobs in order starting from those with the tightest due dates.

Step 4: Align the ordered jobs with the time axis to create a chronological production schedule.

We prepared jobs that simulate a typical electrical discharge machining (EDM) process on the production site, and conducted numerical experiments to derive production schedules by applying the proposed method to them. We describe the experimental method. The prepared jobs were created to apply several test cases, and for each we predetermined an example production schedule. For each test case, we applied the proposed method, compared the resulting schedule with the example schedule, and verified the effectiveness of the proposed method.

The numerical experiments were conducted under the following settings.

(1) Schedule creation period: 7 days
(2) Number of operating machines: 3 units
(3) Working hours per machine per day: 23 hours
(4) Number of experiment runs per test case: 20
(5) Scheduling execution unit: 1 day
(6) Experimental PC CPU: 11th Gen Intel Core*3 i5-1145G7@2.60GHz
(7) Experimental PC memory: 16GB

The scheduling execution unit is the length of the schedule obtained by applying the proposed method once. In these numerical experiments, we set the scheduling execution unit to one day to prevent jobs from spanning across dates.

We prepared three test-case examples as follows and created an example schedule for each. The test cases were designed to increase in difficulty in the order of Case 1, Case 2, and Case 3.

Case 1: Create jobs such that the total processing time for one day is 23 hours. Prepare these for each day in the scheduling period and for each operating machine. At this stage, all jobs have the same priority. In the example, the jobs scheduled on each day are due on that day. Each job's processing time shall also be a multiple of 60 minutes.

Case 2: Assign priorities to each job created in Case 1.

Case 3: For the jobs created in Case 2, add, in a number equal to the operating machines, "500-minute" jobs whose processing time is not divisible by 60 minutes. With the due date of the "500-minute" jobs set to day 7, scheduling even one of them makes it impossible to plan 23 hours of jobs per day without gaps, due-date violations are expected to occur.

As an example for each test case, a one-day schedule example is shown in Fig. 3.

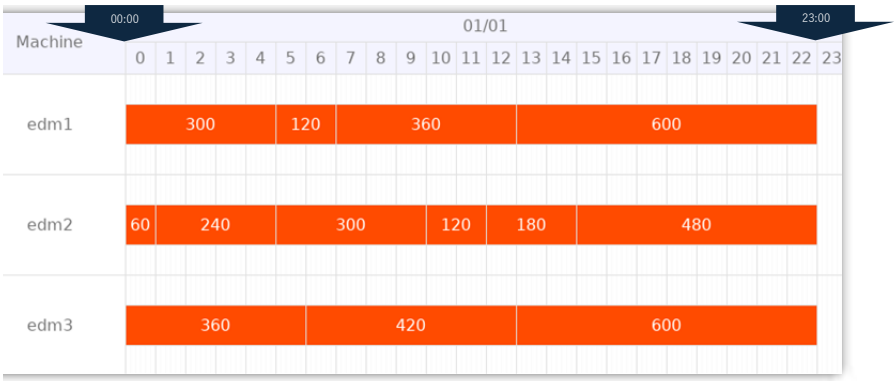


Fig. 3 Ideal example of parallel machine scheduling method

Figure 3 shows a one-day schedule example for three machines. The horizontal axis is the time in one-hour increments, and the vertical axis shows the scheduling results for each machine. From top to bottom: Machine 1, Machine 2, and Machine 3, interrupt jobs are shown in red, and regular jobs in blue. The number inside each job's rectangle also indicates the processing time (minutes). Thus, in the example, a schedule with no idle time from 0:00 to 23:00 can be created. Therefore, if the schedule obtained by the proposed method has no idle time, it can be considered that an optimal solution equivalent to the example has been achieved. If the schedule obtained by the proposed method contains idle time, we evaluate how close it is to the optimal solution by the degree of deviation from the example and define the following evaluation metrics. For every metric, the evaluation indicator is that it is equal to or less than the example.

*3 Intel Core is a registered trademark of Intel Corp.

1) Number of due-date violations: As a result of scheduling each job, if the planned day for processing a job is later than its due date, it is a due-date violation. We use as an evaluation indicator the number of due-date violations obtained by checking this condition for each job.

2) Due-date violation amount: As a result of scheduling each job, if the planned day for processing a job is later than its due date, the job's processing time is counted as the due-date violation amount. However, if the planned day is earlier than the due date, the due-date violation amount is 0. In this case, the sum of the due-date violation amounts across all jobs is used as an evaluation indicator.

3) Computation time: We set an upper limit of 60 seconds and use as an evaluation indicator whether the computation finishes within this limit because the time allowed for scheduling is often limited at production sites.

In formula (11), the total tardiness being minimized is evaluated—among the above evaluation indices—by the number of due-date violations and the due-date violation amount. This is to prevent assigning the same evaluation when either a job with a large processing time or a job with a small processing time becomes tardy, because tardiness is calculated in units of whole days. For both the schedule obtained by the proposed method and the schedule of the assumed example, we evaluate them using these indices and, based on the differences, verify whether the schedule obtained by the proposed method is valid.

For test cases (Cases 1–3), we verify whether the schedules obtained by applying the proposed method satisfy Constraints.1-5. First, by applying the proposed method to each test case, we obtain production schedules. Next, for these schedules, we compute evaluation values based on each evaluation indicator. The evaluation results are shown in Table 1. The values of the evaluation indicators are the averages of data obtained by performing scheduling 20 times.

Table 1 Evaluation results of the proposed method for each index

	Case 1	Case 2	Case 3
Number of due-date violations	0	0	3
Number of correct due-date violations	0	0	3
Due-date violation amount (minutes)	0	0	1500
Due-date correct violation amount (minutes)	0	0	1500
Average computation time (seconds)	1.836	1.970	2.589

Table 1 shows that, across all test cases (Cases 1–3), compared with the assumed example prepared in advance, schedules equivalent in all evaluation indicators were obtained. Also, in the most complex test case, Case 3, Table 1 does not show Constraint.2 Priority constraint —to confirm whether its conditions are satisfied, a diagram of the created schedule is provided in Fig. 4. Figure 4 is an excerpt of the scheduling results by the proposed method, showing Days 7 and 8. If a "500-minute" job is scheduled within the schedule period up to Day 7, the values of the evaluation indicators could be worse than those in the assumed example. However, Fig. 4 results show that they are equivalent to the assumed schedule, and it was confirmed that the values of the evaluation indicators are also equivalent. Moreover, Fig. 4 shows the overall scheduling results for Days 1–8 at the top: red jobs representing interrupt jobs are clustered on Days 1–3, and blue jobs representing regular jobs are clustered from Day 4 onward; therefore, it can be said that Constraint.2 Priority constraint is also satisfied. From these results, we consider that an optimal schedule could be created by formulas (1)–(15). That is, it was confirmed that the proposed method has the basic capability of obtaining an optimal schedule in simple test cases with a small number of jobs[5].

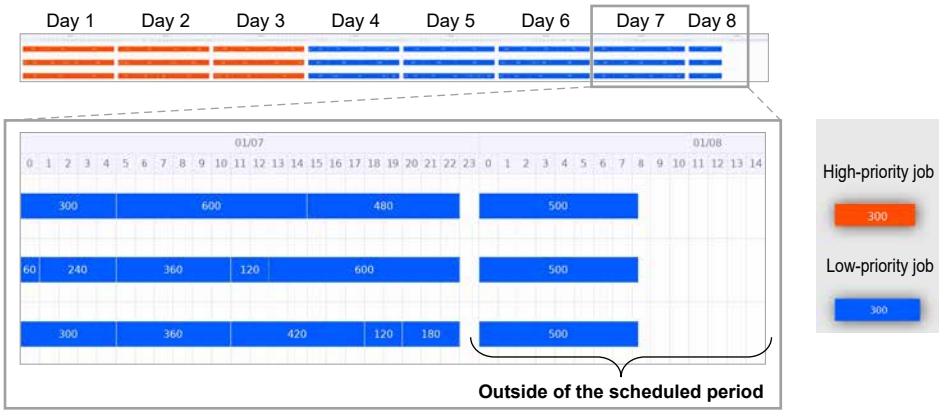


Fig. 4 Result of scheduling at Case 3 (Day 7 to Day 8)

4. Conclusion

At production sites, in situations where factory workers and production equipment coexist, we outlined that: (i) a performance-profile coordinated control method for automating operations by collecting and analyzing line operation data with an edge platform to perform integrated monitoring of the entire line, and (ii) a process scheduling method, characterized by splitting the constraints into an integer-optimization processing part and a rule-based processing part, to rapidly decide the execution order of processes at production sites. Both methods achieved to generate plan in seconds as required in production settings.

References

(1) Council on Competitiveness-Nippon (COCN): New manufacturing with people at the center, Final report of the FY2017 project (2018)
http://www.cocn.jp/report/theme97-L.pdf
(2) Edgecross Consortium
https://www.edgecross.org/ext/ja/index.html
(3) Nakai, A.: A Study on a Method for Linking ROS and Production Systems using an Edge Platform, 2022 Annual Conference on Electronics, Information and Systems, IEEJ, OS1-6 (2022)
(4) Nakai, A.: A Study of Production System Operation Control Method using Profile Information of Workers and Production Equipment, 22nd Forum on Information Technology, J-022 (2023)
(5) Yaguchi, T., et al.: Parallel Machine Scheduling for FA that Considers Processing Priority, 2023 Annual Conference on Electronics, Information and Systems, IEEJ, OS1-2-5 (2023)

A Formal Verification Using Graph-Database for Security Protocols

Authors: *Hisashi Mori*, Kazuki Yonemochi**, Manabu Misawa****

Information Technology R&D Center, ** Itami Works, * Mitsubishi Electric Digital Innovation Corporation*

Abstract

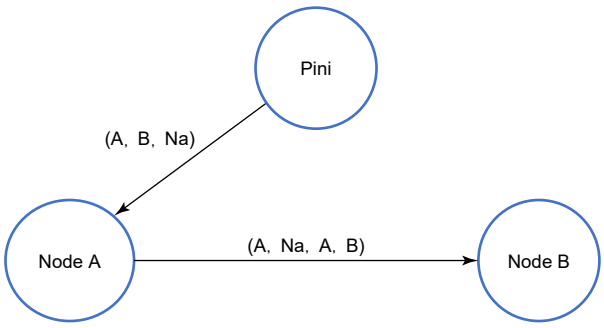
When designing a system, security verification is a method for ensuring that vulnerabilities such as information leakage and unauthorized access are not present. Conventional verification methods require advanced expertise such as formal language theory and tool-specific languages, and suffer from the issue that when systems become complex, verification does not complete within a practical amount of time. Accordingly, rather than using specialized language expressions that can be difficult to immediately understand, we developed a method that enables security verification with data insertion and searching operations by using a general-purpose graph database (hereafter, "graph DB") that provides a visual understanding of relationships among data. As a result of verifying a specific security protocol, intuitive verification with graph visualization detected vulnerabilities similar to those found using conventional techniques. Furthermore, we theoretically demonstrated that verification time can be reduced.

1. Introduction

There is growing demand for ensuring system safety and security. To address such demand, methods are needed that guarantee at the design stage that systems are safe and secure. Security protocols are an example of components that constitute a system, but it is difficult to manually analyze whether a given protocol operates correctly against attacks. To address this problem, automating analysis using verification methods based on formal methods is one effective approach. In verification of security protocols, the protocol and the intruder's behavior with respect to the protocol are modeled in a formal language, the security requirements that the protocol must satisfy are modeled as verification properties in another formal language, and verification is executed using a tool. Such tools include dedicated security protocol verification tools and general-purpose model checkers. Among these, SPIN[1] and other general-purpose model checkers can describe various verification properties, including security, with multiple case studies of security verification existing. Thus, this study focuses on security verification using general-purpose model checkers. A variety of security protocols have been verified with SPIN, and the usefulness of security verification using SPIN has been demonstrated. For example, Maggi et al. conducted verification on the NSPK (Needham-Schroeder Public Key) protocol and derived vulnerabilities[2]. Furthermore, for security protocols in general, Henda logically defined an intruder who performs eavesdropping and insertion as well as retransmission of eavesdropped messages. Based on this definition, verification was conducted on three

```
mtype= {A, B, Na} ;
Chan ca= [0] of {mtype, mtype, mtype, mtype} ;
proctype Plni (mtype self; mtype party; mtype nonce)
{
        mtype g1 ;
        ca ! self, nonce, self, party;
}
proctype PRes (mtype self; mtype nonce)
{
mtype g2, g3;
        ca ? eval (self), g2, g3, eval(self);
}
init
{
        run Plni(A, B, Na)
}
```

(a) Example of a program for SPIN



(b) Example of a graph using a graph-databases

Fig. 1 Examples of describing a target system by the existing method SPIN and the proposed method

types of security protocols, including the NSPK protocol, and known vulnerabilities were derived[3]. However, these conventional techniques require learning specialized languages, making adoption relatively difficult. Furthermore, as the target system increases in complexity, the number of internal states representing the protocol and the intruder's behavior grows exponentially, and verification cannot be completed within a practical timeframe.

In this study, we adopted Neo4j[4]*1, one of the most widely used graph DBs and implemented model checking using a graph DB. Figure 1 shows an example description of the target system using the conventional technique SPIN, and a conceptual diagram representing the same target system as a graph. Figure 1 shows sender A (in the figure, $Node\ A$) sending to receiver B (in the figure, $Node\ B$), the message Na and A, encrypted with receiver B's public key and sent (in the figure, $Node\ A$ to $Node\ B$ arrow) representing this communication. With conventional techniques, after converting the behavior of the target system into a state transition model, that state transition model is programmed in text (Fig. 1 (a)). Whether the program correctly represents the target system needs to be deciphered during verification. Meanwhile, with the graph visualization in the proposed method, the behavior of the system can be intuitively and visually understood (Fig. 1 (b)). Here, $Pini$ denotes the start of the protocol. Also, the characters on the arrow indicate, from left, "sender A, message Na and A, encrypted with receiver B's public key." The burden of conducting security verification can be reduced by being able to understand the target system in an intuitive visual manner.

2. Proposed Method: Security Verification Method Using a Graph Database

We propose a new security verification method that enables intuitive verification and reduces verification time. Section 2.1 describes how, in the proposed method, we define the target system, attack model, and verification properties, which are the input information for the model checking used for this study. Section 2.2 outines the implementation method using a graph DB and the results of actually performing verification.

2.1 Definition of the formal expression of the proposed method

In the proposed approach, we express the security protocol to be verified and the intruder's behavior as a directed graph, and express the properties to be verified as graph DB queries, thereby enabling intuitive verification. In this section, we explain how to define the target system, attack model, and verification properties on the graph DB, and what the verification results indicate for those inputs.

Target system The target system in the proposed approach is a directed graph equivalent to the state transition model of existing model checkers. We express the target system as a directed graph (V, E). Here, V is the set of vertices, E is the set of directed edges. We define the vertices and directed edges as follows.

Vertices are defined to include r (a field indicating the sender/receiver) and s (a field indicating which state in the protocol) as labels. We express the reception of a protocol-conformant message by sender (or receiver) as a single vertex. If the message content differs, each is expressed by a distinct vertex. In addition to those, we add two types of vertices, $Init$ representing protocol start and End representing protocol end.

Directed edges are defined to include t (a field indicating a cipher text or a plain text), m (a field indicating the transmitted message content), k (a field indicating the key that encrypts the message). We regard the communication from the protocol sender to the receiver as a single directed edge.

Attack model The Dolev–Yao model is a well-known attack model that can exhaustively examine vulnerabilities in security protocols, and it is also adopted in Henda's study described in Section 1[3]. Similarly, the proposed approach uses Henda's attack model based on the Dolev–Yao model. That is, we verify whether the security properties are satisfied on the target system augmented with Henda's attack model. For reasons of space, please refer to the original paper[3] for the rigorous definition of Henda's attack model.

We show the procedure for adding the attack model to the target system. First, as preparation, the intruder's set of known information $\mathbb{K}$ is partitioned into the subsets $ACTOR$ (sender/receiver/intruder), $PLAIN$ (plain text), $CIPHER$ (cipher text), $PKEY$ (public key), $SKEY$ (secret key). The procedure for creating the vertices and directed edges that represent attacks when a particular state is used as the source is as follows.

*1 Neo4j is a registered trademark of Neo4j, Inc.

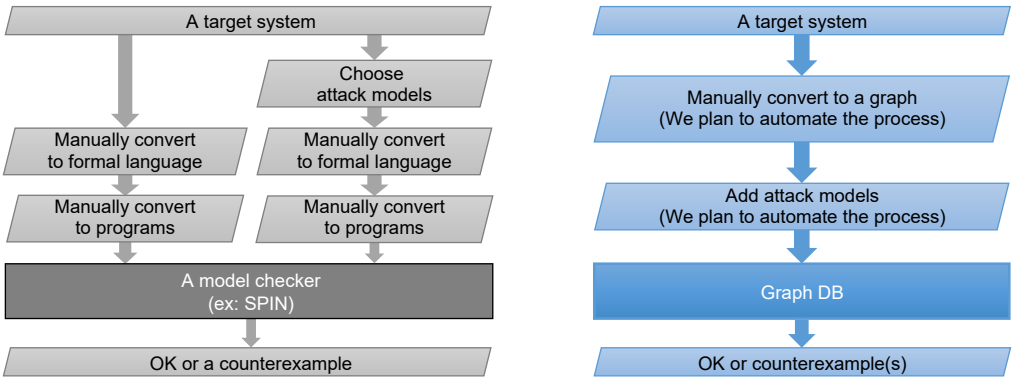
(1) *ACTOR*, *PLAIN*, *PKEY*: select an element from each of them and create cipher texts for all combinations
(2) If the destination vertex (sender or receiver) to send to does not exist, create the destination vertex
(3) Use each cipher text created or the cipher texts already stored in *CIPHER* as labels, and create a directed edge from the source to the destination

Verification property In the proposed approach, as with the conventional technique SPIN, we assume LTL (Linear Temporal Logic) as the formal language for describing verification properties. As in the conventional technique[5], we describe LTL-equivalent formulas in the graph DB. As an example of a verification property, we address user authentication. According to prior work[2] [3], the definition of user authentication in model checking (that sender A indeed authenticates receiver B) is: "Whenever sender A has completed execution of the protocol, the other party B is executing the protocol."

Verification results On the directed graph representing the target system and the attack model, execute the query that represents the verification property; if a path that does not satisfy the query is output, that verification property is not satisfied. As noted earlier, the directed graph and queries in the proposed approach are respectively equivalent to the state transition model and verification properties in conventional techniques (model checkers). Therefore, the output of the queries is equivalent to the verification results of conventional techniques.

2.2 Workflow of model checking using a graph DB

This section describes the workflow of model checking using a graph DB, the proposed method (Fig. 2).



(a) Workflow with the existing methods (b) Workflow with the proposed method

Fig. 2 Workflows for security verification with the existing and the proposed methods

In the conventional technique (Fig. 2 (a)), after selecting the system specifications and attack model to be verified, it was necessary to manually convert them into a formal language, then convert them into the model checker's program, and only then execute verification on the model checker. On the other hand, with the proposed method (Fig. 2 (b)), the system specifications and attack model to be verified are selected, and the verification simply executed to obtain the results. The proposed method is an algorithm that converts the protocol and attack model into a graph DB for any security protocol. Once the algorithm is established, automating the conversion is readily possible. Going forward, we plan to automate the conversion of protocols and attack models into the graph DB. Using a graph DB in the model checking method enables designers to perform model checking with knowledge of ordinary database operations—adding vertices and directed edges, and performing path searches. Also, by using a graph DB and appropriately reducing the number of states of the intruder, the variables representing information sent and received in the protocol, and forged messages, verification using the proposed method can reduce the number of states required for verification compared with the conventional techniques.

As an example demonstrating gragh DB being implemented, we targeted the NSPK protocol mentioned in Chapter 1 and verified authentication, the verification property described in Section 2.1. Specifically, we expressed the NSPK protocol, the attack model, and the logical formula expressing authentication on the graph DB following the procedure shown in Section 2.1, and executed both a method to find the shortest path among the paths that do not satisfy authentication and a method to find all paths. Here, when the verification target and the verification property is input and the verification is run, if the verification target

does not satisfy that verification property, paths in the verification target that do not satisfy the verification property are searched and output. Conversely, if the verification target satisfies the verification property, nothing is output. As a result of running the verificatoin, counterexamples were output by both methods. By visually confirming that the two types of paths do not satisfy the verification property, we verified that the results are correct. Note that, due to space limitations, the graphs expressing the NSPK protocol and the verification results are omitted.

3. Assessment

Both the verification method using SPIN as the conventional technique and the verification method using a graph DB in the proposed method create a model of the behavior of the target system as a state transition model, and output a counterexample when a state that does not satisfy the verification property is reached. That is, the verification time depends on the number of states generated. If there are too many states, there is a risk that the verification will not complete in a practical amount of time. Therefore, in this chapter, as a logical evaluation of execution time, we compare the number of states generated by the conventional technique and by the proposed method. As a result of comparing the number of states, we confirmed that the proposed method always yields fewer states than the conventional technique. For example, for the NSPK protocol (number of messages m = 3), the number of states in the conventional technique is 15,275, while the proposed method has 20 states, which is about 1/763 (Fig. 3). The number of states generated by the conventional technique SPIN is the product of the number of states of the target system and the number of states for the verification property. By contrast, the number of states in the proposed method is just the number of states of the target system. This is because, thanks to the graph DB's search functionality, verification can be performed by searching only states of the target system. The results of calculating and comparing the number of states for the conventional technique and the proposed method, respectively are showing as follows.

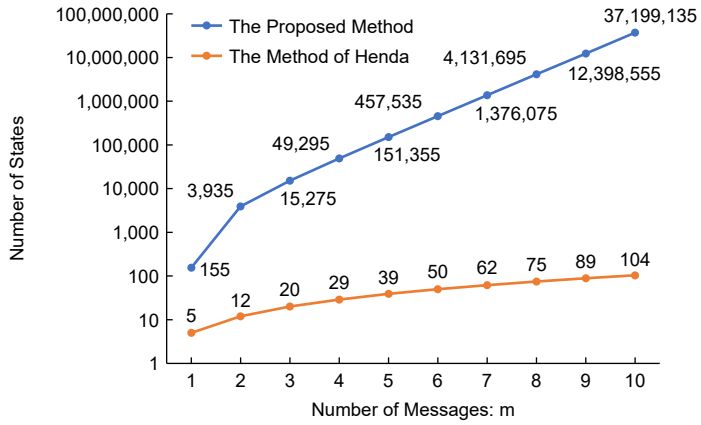


Fig. 3 Comparison of the number of states for the existing and the proposed methods

In Henda's conventional approach, the number of states is given by the following equation.

$$\{m+3+\sum_{i=0}^{m}\{(|Knows|-|PKEY|)^{(t-t_{|PKEY|}+(i-1))} \times\{(|Knows|-|PKEY|)^{(t-t_{PKEY})}\times|PKEY|+1\}\}\} \times(2^{|TL|}+1)$$

Here, m is the number of messages; $|Knows|$ is the number of elements of the information known by the intruder; $|PKEY|$ is the number of elements among the intruder's known information that are public keys; t is the number of terms per message; $t_{|PKEY|}$ is the number of terms per message that represent public keys; $|TL|$ is, if we define TL as the set of formulas in which the outermost operator of each subformula of the verification property is a modal operator, the number of elements of TL.

Meanwhile, in the proposed approach, the number of states is given by the following formula.

$$(m+3)+|Init|+\sum_{i=0}^{m}((|PLAIN|^{t_{PLAIN}}+i+1))$$

Here, | $Init$ | is the number of states of the initial sender; | $PLAIN$ | is the number of elements in the entire plain text; t_{PLAIN} is the number of terms per message that express plain text.

Based on the above formula, for Henda's method and the proposed approach, a graph comparing the change in the number of states with respect to the number of messages m is shown in Fig. 3. Overall, the proposed approach results in a smaller number of states. On a typical PC, the verification limit is said to be around 500,000 states[6]. With the conventional approach, when the number of messages is 7, the number of states exceeds 500,000, whereas with the proposed approach it is kept at 62.

4. Conclusion

Using a security protocol as an example, we proposed a model checking method that describes the target system as a directed graph in a graph DB and specifies the verification property as a query in the graph DB. Using general graph DB knowledge, one can create graphs, and visualization functions allow for an intuitive understanding. In addition, by using the graph DB's all-path search function, it is also possible to execute model checking that outputs all counterexamples. Furthermore, by directly specifying the verification property as a query over the directed graph and searching, the number of states during model checking can be reduced, making automated verification feasible within a practical time.

Going forward, when this is applied to other security protocols, we plan to evaluate how much the execution results and the number of states differ between the conventional technique SPIN and the proposed approach. Furthermore, we plan to show that the proposed approach can also define and verify properties other than authentication. We also plan to extend the formalization of the target system and verification property so that verification including system safety can be performed. Going forward, systems will become increasingly complex, making it difficult to manually determine whether security functions are correctly implemented. By using the proposed approach to automatically assure system security, the workload in the design phase can be significantly reduced.

References

(1) Holzmann, G.J.: The SPIN Model Checker, Addison-Wesley (2004)
(2) Maggi, P., et al.: Using SPIN to Verify Security Properties of Cryptographic Protocols, Proceedings of the 9th International SPIN Workshops on Model Checking on Software, LNCS 2318, 187–204 (2002)
(3) Henda, B,N.: Generic and Efficient Attacker Models in SPIN, Proceedings of the 2014 International SPIN Symposium on Model Checking of Software, 77–86 (2014)
(4) Robinson, I., et al.: Graph Databases, 2nd Edition, O'Reilly Media (2015)
(5) Kuno, K., et al.: A Model Checking Approach Using Graph Database, Information Processing Society of Japan Technical Report (SE), 2018-SE-198 (2018)
(6) Jøsang, A.: Security Protocol Verification using SPIN, the first SPIN Workshop (SPIN'95), 1–9 (1995)

Insider Threat Detection Using Decoy

Authors: *Takumi Yamamoto*, Kohei Nozawa**

**Information Technology R&D Center*

Abstract

Information leakage due to cyberattacks is often cited as a major issue, but in reality numerous information leakage incidents caused by insiders also occur[1]. An insider is a malicious user with legitimate access privileges, and it is difficult for existing security measures to properly detect such signs. We focused on decoys as a means of uncovering the malicious intent of insiders. We developed a method that dynamically places decoy files for users whose activities differ from usual and narrows down insiders based on access patterns to those decoy files.

1. Introduction

"Damage from information and other leakages due to malicious insiders" ranks fourth in IPA (Information-technology Promotion Agency, Japan)'s "Top 10 Information Security Threats 2025" and has been treated as a top ten threat for ten consecutive years[2]. In OT (Operational Technology) environments such as critical infrastructure and factories, malicious activity by insiders is also recognized as a serious threat[3]. It follows that measures against insiders need to be considered. This paper describes Mitsubishi Electric's efforts on technical measures against insiders.

2. Insider Threats

In this paper, an insider is defined as a malicious user who has a legitimate account. Given that traditional monitoring and defense models generally trust users once they have been authenticated, it is difficult to counter insiders using those models. Therefore, applying behavior analysis technologies such as anomaly detection and User and Entity Behavior Analytics (UEBA), which evaluate deviations from users' normal behavior—analyzed in advance—as a risk value and exposes high-risk users, is considered effective[4]. However, users' job duties vary widely, and even when the same user performs the same task, there are variations in users' behavior; therefore, it is difficult to calculate the risk value accurately. Insiders may also commit malicious activities within the range that behavior analysis technologies evaluate as low risk in order to avoid detection (for example, uploading confidential files in small quanities), and strict monitoring with behavior analysis technologies can lead to frequent false alarms.

For these reasons, accurately identifying insiders is considered to be a very difficult challenge. Given that they carry out malicious acts within the scope of legitimate access privileges, the difference between insiders and users without malicious intent (referred to as "legitimate users") is only the presence or absence of malicious intent, and merely passively observing behavior makes it difficult to distinguish them.

3. Proposed Insider Countermeasure Technology

In light of this, rather than separating insiders from legitimate users solely based on passively observable information, we adopted an approach in which the defending side actively induces actions that an insider would likely take, indirectly uncovering the insider's malicious intent and observing actions related to that intent. To indirectly uncover an insider's malicious intent, the concept of the proposed method is to dynamically place decoy files for users whose activities differ from usual—files likely to attract interest if the user is an insider—and to narrow down insiders based on access patterns to the decoy files.

An overview of the proposed method is as follows (Fig. 1).

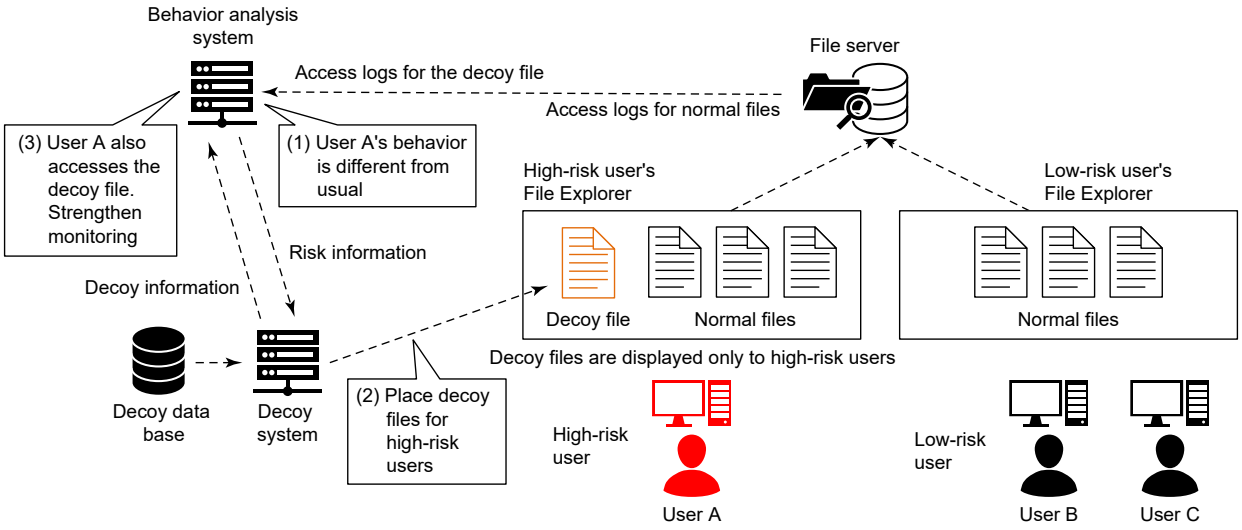


Fig. 1 Overview of the proposed method

(1) Using a behavior analysis system such as anomaly detection, analyze file access logs, etc., and calculate anomalies in behavior as a risk value. Furthermore, create risk information that includes the target user name, risk value, and file access information.

(2) The decoy system receives risk information from the behavior analysis system and, for high-risk users, infers from file access information the topics likely to attract the interest of the user, then selects decoy files containing content related to those topics from a decoy data base in which decoy files on various topics are stored. After that, control the display of the user's file explorer so that the selected decoy file appears as if it is placed on the file server. By placing decoy files that are appealing to insiders, the probability of selecting the decoy files increases, making insiders easier to identify. Information about the placed decoy files (file names, target user names, etc.) is passed to the behavior analysis system as decoy information.

(3) The behavior analysis system also monitors access to decoy files, just like normal files. The behavior analysis system determines whether there has been access to decoy files based on decoy information from the decoy system. If access to a decoy file occurs, this is fed back to the behavior analysis system's risk value (or threshold) to strengthen monitoring of higher-risk users who access decoy files in a proactive manner and narrow them down as potential insiders.

By limiting the placement of decoy files to users determined to be high risk, we can reduce disruption to the work of low-risk legitimate users performing their usual tasks. In the unlikely event that a legitimate user is judged high risk and decoy files are placed, it is expected that they will not actively access files other than those they usually use, so the likelihood of business disruption is low.

4. Prototype and Operational Image of the Proposed Approach

We created a prototype to verify the feasibility of the proposed approach. The configuration of the prototyped system is shown in Fig. 2.

Details are available in our existing research paper[5], however, Fig. 2 's decoy selection server, decoy distribution server, decoy displaying plugin, and decoy data base work together, and the functions corresponding to the decoy system in Fig. 1 are achieved. Given the recent increase in cloud-based file sharing, in this prototype Fig. 1 's file server was not used; instead, on SharePoint[6] we placed content that simulates files used for business. The behavior analysis system was built on Azure[7] using Elastic Stack[8]. The decoy distribution server was also built on Azure. The decoy selection server, which obtains reference information for decoy files from the decoy data base, was built in the business environment. The decoy data base was placed on the same SharePoint as the simulated business content. Assuming access to files on SharePoint via a browser, the control to make only target users able to view decoy files was implemented by installing a decoy displaying plugin in the browser.

4.1 Flow of operation of the prototype system

The following outlines the flow of operations in the system configuration (Fig. 2).

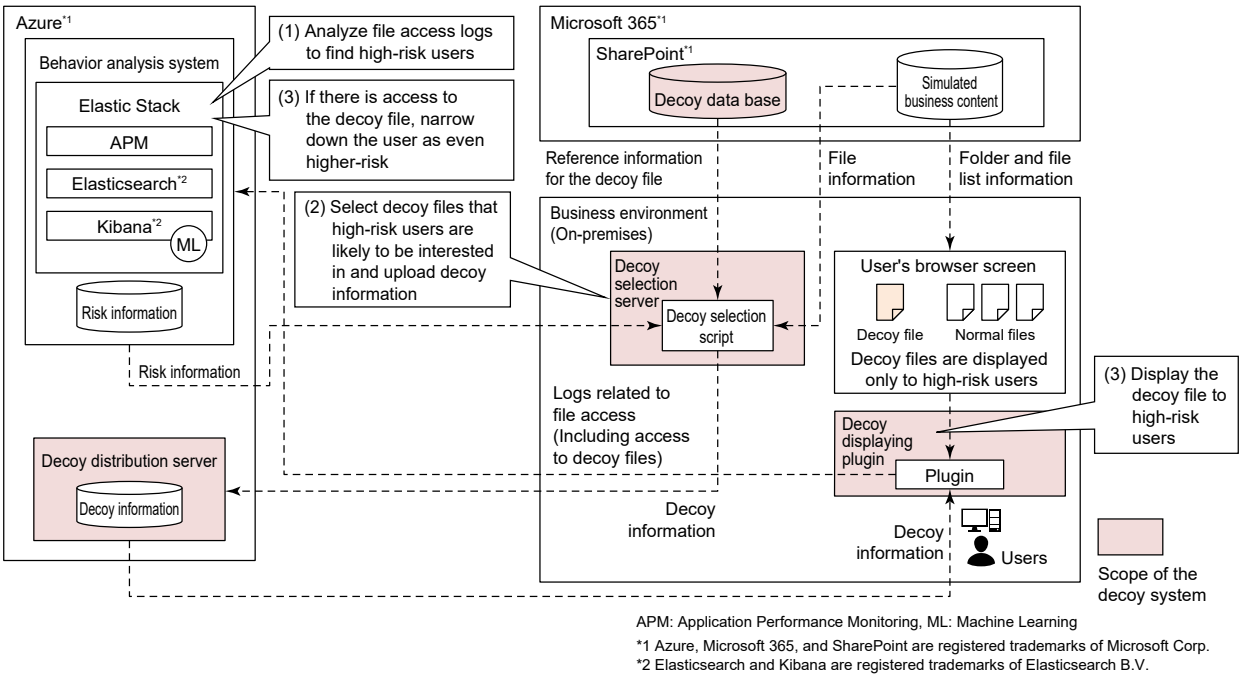


Fig. 2 Configuration of the prototype system

(1) The behavior analysis system analyzes logs related to file access on SharePoint and calculates a risk value. Furthermore, it creates risk information that includes the names of high-risk users, the risk value, and information on file accesses that were determined to be high risk.

(2) The decoy selection server periodically obtains the risk information from the behavior analysis system and estimates file topics from information about the files that high-risk users accessed most recently (such as file names and file contents). After obtaining reference information for decoy files related to the topic from the decoy data base, it creates decoy information for the relevant user and uploads it to the decoy distribution server.

(3) The decoy displaying plugin periodically obtains decoy information from the decoy distribution server and displays decoy files if the user is high risk. In this system, instead of the decoy information (Fig. 1) that is sent from the decoy system to the behavior analysis system, the decoy displaying plugin sends to the behavior analysis system, together with the logs of accesses to files on SharePoint, information (labels) indicating whether a decoy file was accessed. When the behavior analysis system confirms access to a decoy file, it further narrows down the user as higher risk.

4.2 Operational image of the prototype system

Using the dashboard of risk values from the behavior analysis and the screen where decoy files are displayed as examples, the following outlines the operational image of the prototype (Fig. 3).

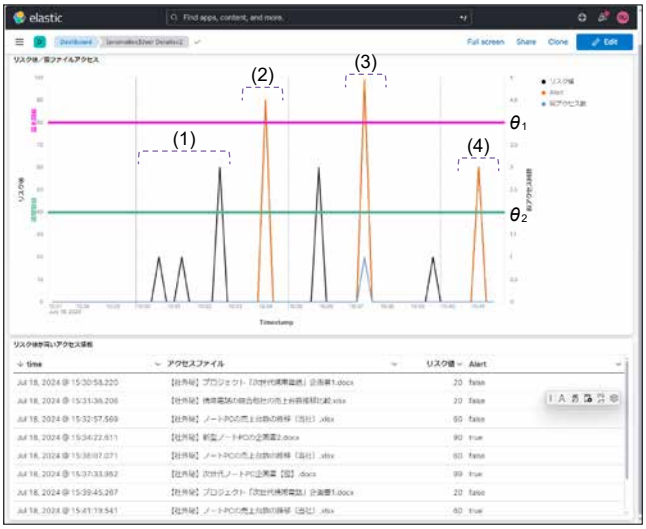
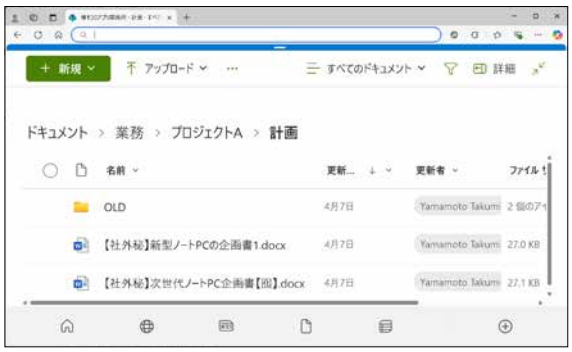


Fig. 3 Example of a dashboard for risk values in behavior analysis

(1) The behavior analysis system analyzes logs related to the user's file accesses on SharePoint and calculates a risk value. An example of visualizing risk values on a dashboard is shown in Fig. 3. In the graph, the horizontal axis is time and the vertical axis is the risk value, and the line shows the calculated risk values for a given user's behavior at each point in time.

(2) The behavior analysis system generates alerts when the risk value exceeds the specified threshold, and creates risk information. The decoy selection server verifies the presence of an alert by retrieving the risk information. Under normal conditions, the threshold is set to θ_1 (pink line in Fig. 3). In the example in Fig. 3, at point (2), the risk value exceeded the threshold θ_1, so risk information is created immediately afterward and the decoy selection server confirms the alert. In addition to the risk value, alerts include items such as the threshold and information about the accessed file (bottom of dashboard in Fig. 3).

(3) When an alert is generated, a decoy file is displayed to the relevant user. Based on information obtained from the decoy data base, it is also possible to reproduce the decoy file's meta information (timestamp, file size, last updater, etc.). An example of displaying a decoy file in a browser is shown in Fig. 4. In the example in Fig. 4 (a), the string "囮" (meaning "decoy") is added to the file name for explanatory purposes, so it is clear whether it is a decoy file. The example in Fig. 4 (b) shows the screen when a decoy file is opened in a browser. Note that, in the figures in this paper, the file names and file contents shown are entirely fictitious and have no relation to actual business operations. Also, while the decoy files were prepared manually this time, we believe that by leveraging generative AI technologies that are evolving rapidly in recent years, it would be possible to automatically generate large quantities of high-quality decoy file content; however, careful attention is required regarding the copyrights of the generated content.



(a) Decoy file displayed on the SharePoint screen

(b) Screen when opening a decoy file in a browser

Fig. 4 Example of a displayed decoy file

(4) When a user accesses a decoy file, the alert generation threshold is changed ($\theta_1 \rightarrow \theta_2$) to a more stringent value for that user only and monitoring is increased. In the example in Fig. 3, the threshold used changes from the pink line (θ_1) to the green line (θ_2), so that alerts will be generated even for lower risk values (in Fig. 3, (4)). If there is no access to a decoy file for a certain period, the threshold will be returned to its original value (θ_1). To make it clear whether a decoy file was accessed, when there has been access to a decoy file, a blue line graph is drawn in addition to the risk value line graph (Fig. 3, (3)). Whether a file is a decoy file is determined based on the label attached to the file access log.

Note that, for explanatory purposes, the risk values in Fig. 3 are calculated based on predetermined rules. In actual operation, it is also possible to use machine-learning-based anomaly detection technologies that do not require manually defining rules in advance.

4.3 Addressing the possibility of the countermeasure being disabled

Finally, it is conceivable that an insider who learns that this countermeasure has been introduced may try to disable it. For attacks that attempt to disable the plugin, they can be mitigated by enforcing the plugin via the organization's group policy and configuring the environment so that SharePoint hosting confidential files cannot be accessed without the plugin. For insiders who try to distinguish decoy files by checking file meta information or decoy information, it is possible to counter them by scoring such actions as dangerous behavior and applying additional monitoring.

5. Conclusion

To protect against insider threats that have been difficult to detect using conventional methods, we proposed a method that indirectly uncovers an insider's malicious intent by dynamically placing decoy files likely to attract the interest of users who behave differently than usual, and narrowing down insiders based on tendencies of accessing the decoy files. We also presented a simple example of implementation and demonstrated its feasibility.

Going forward, we will conduct user studies based on past insider cases and examine the effectiveness of the proposed approach.

References

(1) Proofpoint, Inc.: 2022 COST OF INSIDER THREATS GLOBAL REPORT (2022)
https://protectera.com.au/wp-content/uploads/2022/03/The-Cost-of-Insider-Threats-2022-Global-Report.pdf
(2) Independent Administrative Agency Information-technology Promotion Agency: "Top 10 Information Security Threats 2025" (2025)
https://www.ipa.go.jp/security/10threats/10threats2025.html
(3) National Counterintelligence and Security Center: INSIDER THREAT MITIGATION FOR U.S. CRITICAL INFRASTRUCTURE ENTITIES, GUIDELINES FROM AN INTELLIGENCE PERSPECTIVE (2024)
https://www.dni.gov/files/NCSC/documents/nittf/20240926_Insider-Threat-Mitigation-for-US-Critical-Infrastructure.pdf
(4) IBM Japan, Ltd.: What is User and Entity Behavior Analytics (UEBA)
https://www.ibm.com/jp-ja/topics/ueba
(5) Yamamoto, T., et al.: Proposal of an Insider Detection System Using a Decoy, Computer Security Symposium 2024 Papers, 494–500 (2024)
(6) Microsoft Corp.: What is SharePoint?
https://support.microsoft.com/en-us/office/what-is-SharePoint-97b915e6-651b-43b2-827d-fb25777f446f
(7) Microsoft Corp.: What is Azure?
https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure
(8) Elasticsearch B.V.: Elastic Stack
https://www.elastic.co/jp/elastic-stack

Opto-electronic Convergence Technology for Achieving High Capacity and Low Power Consumption in Next-generation Data Centers

Authors: *Nobuo Ohata*, Mizuki Shirao**

**Information Technology R&D Center*

Abstract

With the emergence of generative AI represented by Large Language Models (LLM) in recent years, the amount of computation required for machine learning has continued to grow immensely. As computation capacity increases, the power consumption of AI data centers continues to increase, and has become a major social issue. As one solution to this challenge, opto-electronic convergence technology has been attracting attention. Co-packaged optics, which compactly integrate electrical ICs and optical devices, can deliver high-capacity communications and lower power consumption that are difficult to achieve with conventional optical transceivers, and are expected to be deployed in next-generation AI data centers. After outlining the evolution of backend networks in GPU clusters and the features of co-packaged optics, this report introduces our initiatives toward co-packaged optics.

1. Introduction

With the emergence of generative AI represented by LLM, the computation capacity required for machine learning has continued to grow immensely. Figure 1 plots the normalized computational performance (FLOPS) of processors, the network bandwidth for computing, and the number of parameters used in machine learning[(1)]. The number of parameters used in machine learning has been increasing at a pace of about 600-fold every two years; OpenAI's GPT-3[*1] uses 175 billion parameters, and Switch Transformer and GPT-4[*1] use over 1 trillion parameters. Meanwhile, the computational performance of processors increases about 2.6 times every two years, and the network bandwidth required for computing increases about 1.5 times every two years; improvements in computing performance are modest compared with the growth in parameters. Therefore, by interconnecting more than several thousand GPUs in parallel via optical communications and clustering them, computational performance is greatly enhanced for executing machine learning. However, the computations require many processors and optical communication equipment, leading to enormous power requirements at AI data centers. To address this challenge, we have begun exploring co-packaged optics that can balance power efficiency in optical communications with expanded network bandwidth. This report presents trends in backend networks required for machine learning and technological trends in co-packaged optics configurations. After introducing the characteristics of the high-speed Electro-absorption Modulator integrated Laser (EML) we have developed to date, we describe high-density integration technologies for applying them to co-packaged optics.

*1 GPT-3 and GPT-4 are registered trademarks of OpenAI OpCo, LLC.

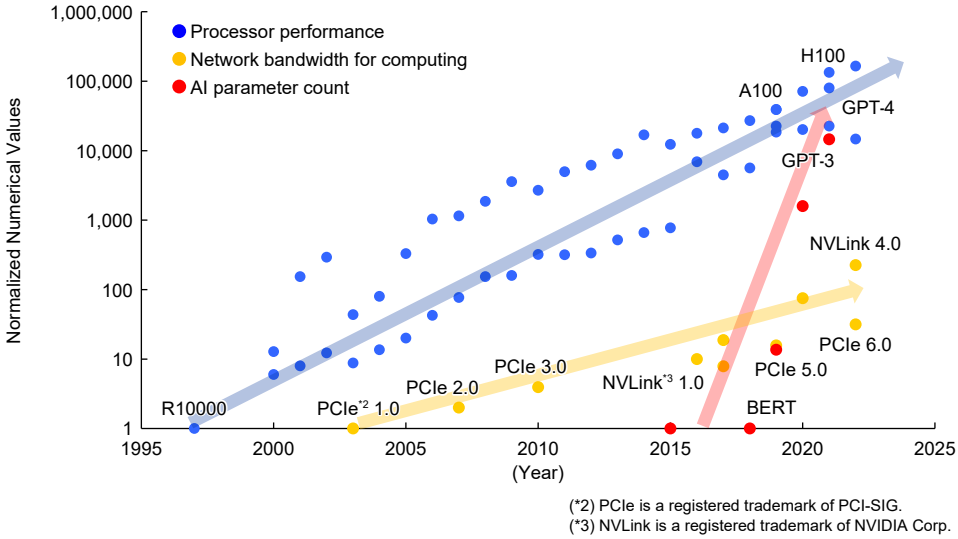


Fig. 1 Trends in processor performance, network bandwidth, and AI model parameters

2. Network Trends in AI Machine Learning and the Introduction of Co-packaged Optics

Figure 2 shows a conceptual diagram of the backend network. The configuration connects many GPUs in parallel through multiple switches. At NVIDIA, 72 GPUs are interconnected in parallel via NVSwitch using a proprietary high-speed network called NVLINK(2), but going forward, to enhance computational capacity, scaling is expected to progress from the current 72 GPUs to 576 and 1,152. It is expected that the number of optical transceivers used to connect GPUs and switches via optical communications will also increase, raising the issue of significantly higher power consumption. Furthermore, as the number of connections to GPUs increases, switch Application-Specific Integrated Circuits (ASIC) will also grow in capacity, and the signal bandwidth output from the switch ASIC will increase accordingly. Therefore, optical transceivers will need wider communication bandwidth, and the amount of signals that can be drawn per unit length—that is, edge density (Tb/s/mm)—will become an important metric going forward. Figure 3 shows the results of calculating edge density, referencing the trends shown in Fig. 1. Edge density will need to reach 1Tb/s/mm by 2028 and will increase exponentially to 2Tb/s/mm by 2030.

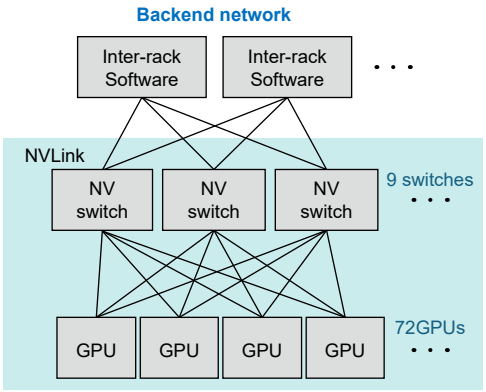


Fig. 2 Schematic diagram of the backend network

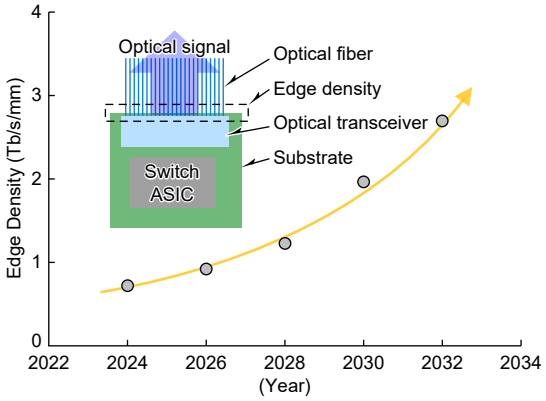
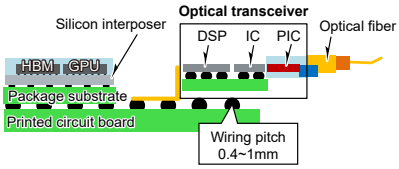
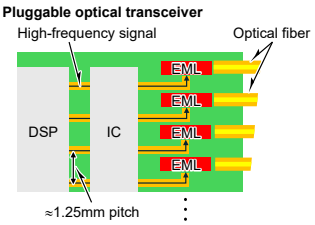
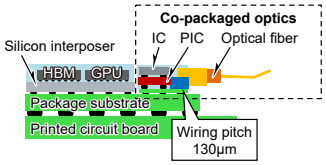
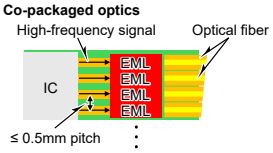


Fig. 3 Trends in bandwidth edge density

Because existing pluggable optical transceivers find it difficult to achieve low power and high edge density, co-packaged optics are expected to be a new technology that can solve these challenges. Table 1 compares pluggable optical transceivers and co-packaged optics. A pluggable optical transceiver is used by connecting to an electrical connector mounted on a printed circuit board. Therefore, the distance from logic ICs such as GPUs and switch ASIC to the transceiver is lengthy, requiring a DSP (Digital Signal Processor) to correct degraded electrical signals, which leads to high power consumption. Power efficiency is approximately 20pj/bit. Also, edge density is 0.1Tb/s/mm or lower due to wiring pitch constraints imposed by the connector. In contrast, co-packaged optics are mounted on the package substrate on which the logic

IC is installed, eliminating the need for a DSP and enabling power efficiency of 5pj/bit. Moreover, because they are implemented on substrates that support fine wiring, edge density of up to 2Tb/s/mm is achievable. Furthermore, in the future, configurations that mount on a silicon interposer are being considered; in that case, on the electrical L/S (line-and-space), edge density exceeding 2Tb/s/mm would be possible.

Table 1 Comparison of pluggable optical transceivers and co-packaged optics

	Implementation Diagram	Transceiver Top View	With or Without DSP	Power Efficiency	Bandwidth Edge Density
Pluggable optical transceivers			With	Approx. 20pj/bit	<0.1 Tb/s/mm
Co-packaged optics			Without	Approx. 5pj/bit	<2 Tb/s/mm

We have begun examining co-packaged optics incorporating EML. An EML is a laser device that integrates a Distributed Feedback Laser Diode (DFB-LD) and an Electro-absorption Modulator (EAM), which absorbs light when a voltage is applied, on a single chip. Compared with Si photonics modulators, which excel at high-density integration, EML enables higher-speed operation at lower power, making it possible to achieve higher edge density. However, in co-packaged optics, EML must be placed in high densities, and it is difficult to achieve this with conventional assembly that connects the high-frequency traces to the EML with wires. We therefore considered adopting flip-chip bonding.

3. EML Implementation Technology for Low Power and High Edge Density

In this chapter, we introduce the ultra-high-speed EML[3][4] that we have developed, and describe EML implementation forms applicable to co-packaged optics and their characteristics.

3.1 Implementation configuration

Our EAM employs a unique high-mesa waveguide structure in which the optical absorption layer is sandwiched on both sides by low refractive index material (Fig. 4). This structure strongly confines light in the high-refractive-index optical absorption layer, enabling efficient optical absorption. In other words, sufficient optical absorption is achieved even with an EAM of low capacitance, enabling high-speed operation. Given that the EAM is located in front of the EML (on the light-output side), it is common to route high-frequency traces along the side of the EML and use gold wire for wire bonding to the EAM electrodes (Fig. 5). Wire bonding requires a large mounting footprint; in past developments it occupied a width of about 1.25mm per EML, limiting edge density. To achieve the high edge density required for co-packaged optics, we developed an implementation technology that does not use wire bonding.

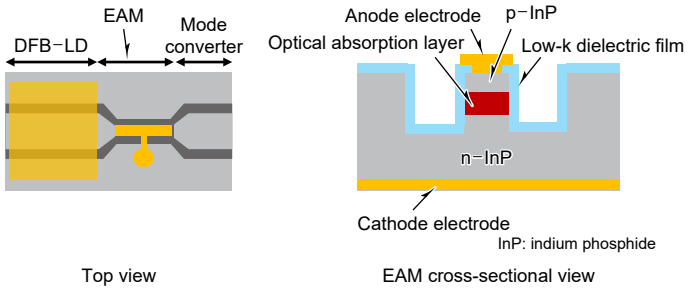


Fig. 4 EML employing high-mesa EAM

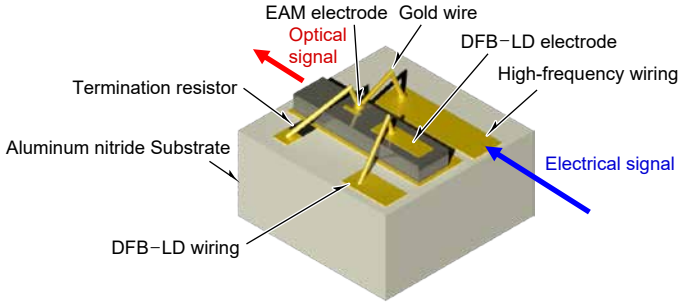


Fig. 5 Conventional assembly using wire bonding

3.2 Performance assessment

Figure 6 shows the configuration of the co-packaged optics. In co-packaged optics, the electronic IC and optical devices are mounted using flip-chip technology. In flip-chip bonding, the electrodes are directly joined using bumps, etc., so wire bonding as shown in Fig. 5 is not used. As a result, not only is a substantial increase in assembly density possible compared with conventional structures, but the parasitic inductance from wire bonding is eliminated, enabling higher operating speed. Simulations for flip-chip bonding indicate an operating speed of 145 GHz (Fig. 7).

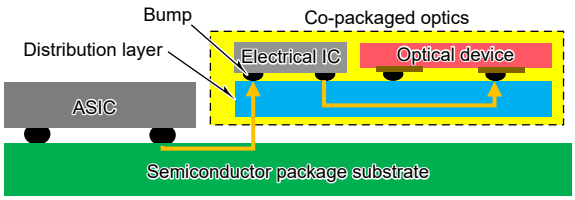


Fig. 6 Cross-sectional view of co-packaged optics using EMLs

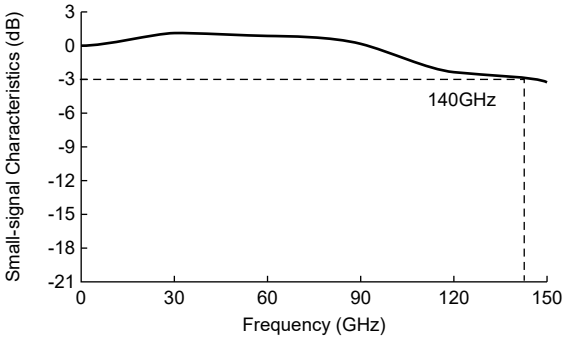


Fig. 7 Simulation results of co-packaged optics using EMLs

Experimental verification of flip-chip bonding onto an aluminum nitride substrate achieved an operating speed of 90GHz[5]. The small-signal characteristics are shown in Fig. 8. Ripples not seen in Fig. 7 are observed in the small-signal characteristics; these are believed to be due to impedance mismatches in the high-frequency interconnects. Going forward, if advances in manufacturing enable the arraying of EML, densification down to around 0.25mm—five times higher than before—can be expected.

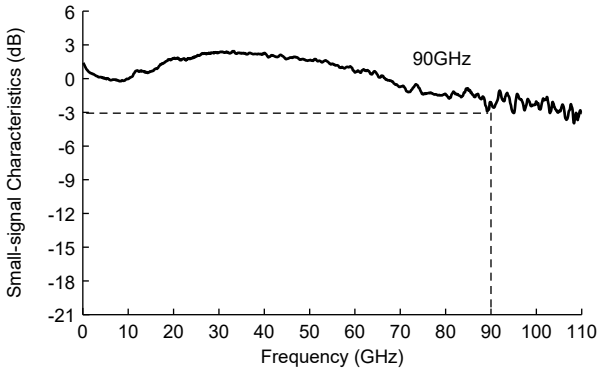


Fig. 8 Small-signal response of EML in co-packaged optics[5]

3.3 Future outlook

To increase the capacity of next-generation data centers, it will be necessary to optimize the device structure—including reducing EAM capacitance and arraying—to improve operating speed. In addition, using low-dielectric-constant substrate materials (e.g., quartz substrates) is expected to further improve operating speed. By achieving these and applying the next-generation modulation format, 200 Gbaud PAM4 (Pulse-Amplitude Modulation, 4 levels), an edge density of 1.6Tb/s/mm is anticipated.

4. Conclusion

This report presented trends in backend networks and co-packaged optics, introduced EML, and described our efforts toward co-packaged optics. Flip-chip bonding technology using EML with a high-mesa structure enables the high edge density required for co-packaged optics. With advances in AI and machine learning, the importance of opto-electronic convergence technology is expected to grow further, and, in addition to advances in optical device technology, advances in the packaging technologies for optoelectronics will be essential. We will continue development toward next-generation opto-electronic convergence technology.

References

(1) riselab: AI and Memory Wall
https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8
(2) NVIDIA: NVLink and NVSwitch
https://www.nvidia.com/ja-jp/data-center/nvlink/
(3) Uchiyama, A., et al. : Demonstration of 155-Gbaud PAM4 and PAM6 Using a Narrow High-Mesa Electro-Absorption Modulator Integrated Laser for 400Gb/s Per Lane Transmission, Journal of Lightwave Technology, 43, No.4, 1868-1873 (2025)
(4) Shirao, M., et al. : High Speed EML and Assembly Techniques for GPU Cluster System, 2025 Optical Fiber Communications Conference and Exhibition, 1-3 (2025)
(5) Masuyama, K., et al. : EML Assembled with Flip-Chip Technology on AlN Sub-mount Operating at 212.5Gbps PAM4, OECC 2025, WG2-2 (2025)

MITSUBISHI ELECTRIC CORPORATION