I see an IC: A Mixed-Methods Approach to Study Human Problem-Solving Processes in Hardware Reverse Engineering

René Walendy*
Ruhr University Bochum
Bochum, Germany
rene.walendy@rub.de

Markus Weber
Ruhr University Bochum
Bochum, Germany
markus.weber3@rub.de

Jingjie Li
University of Edinburgh
Edinburgh, United Kingdom
jingjie.li@ed.ac.uk

Steffen Becker*
Ruhr University Bochum
Bochum, Germany
steffen.becker@rub.de

Carina Wiesen
Ruhr University Bochum
Bochum, Germany
carina.wiesen@rub.de

Malte Elson
University of Bern
Bern, Switzerland
malte.elson@unibe.ch

Younghyun Kim
University of Wisconsin–Madison
Madison, Wisconsin, United States
yhkim1@purdue.edu

Kassem Fawaz
University of Wisconsin–Madison
Madison, Wisconsin, United States
kfawaz@wisc.edu

Nikol Rummel
Ruhr University Bochum
Bochum, Germany
nikol.rummel@rub.de

Christof Paar
Max Planck Institute for Security and Privacy, Germany
christof.paar@mpi-sp.org

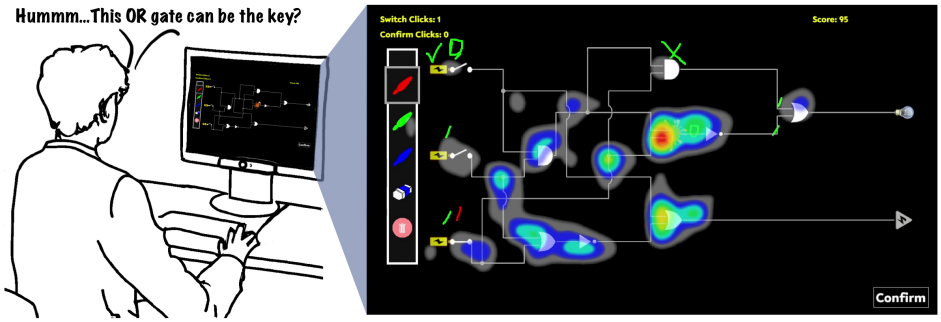


Figure 1: An illustration of a participant solving hardware reverse engineering tasks while thinking aloud, with eye tracking recorded, along with a heatmap of their visual attention (eye fixations) on the screen.

ABSTRACT

Trust in digital systems depends on secure hardware, often assured through Hardware Reverse Engineering (HRE). This work develops methods for investigating human problem-solving processes in HRE, an underexplored yet critical aspect. Since reverse engineers rely heavily on visual information, eye tracking holds promise for studying their cognitive processes. To gain further insights, we additionally employ verbal thought protocols during and immediately after HRE tasks: Concurrent and Retrospective Think Aloud. We evaluate the combination of eye tracking and Think Aloud with 41 participants in an HRE simulation. Eye tracking accurately identifies fixations on individual circuit elements and highlights critical components. Based on two use cases, we demonstrate that eye tracking and Think Aloud can complement each other to improve data quality. Our methodological insights can inform future studies in HRE, a specific setting of human-computer interaction, and in other problem-solving settings involving misleading or missing information.

*Also with Max Planck Institute for Security and Privacy.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642837

CCS CONCEPTS

• **Security and privacy** → **Hardware reverse engineering**; • **Hardware** → **Logic circuits**; • **Human-centered computing** → **Empirical studies in HCI**; *Laboratory experiments*; *User studies*.

KEYWORDS

hardware reverse engineering, integrated circuits, eye tracking, think aloud, mixed-methods, problem solving, semiconductor industry

ACM Reference Format:

René Walendy, Markus Weber, Jingjie Li, Steffen Becker, Carina Wiesen, Malte Elson, Younghyun Kim, Kassem Fawaz, Nikol Rummel, and Christof Paar. 2024. I see an IC: A Mixed-Methods Approach to Study Human Problem-Solving Processes in Hardware Reverse Engineering. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3613904.3642837

1 INTRODUCTION

With a strong reliance of society on information technology, people need to be able to trust an increasing variety of digital systems in their everyday lives. To build systems that can be truly trusted, we must be able to provide high assurance not only for all software layers but also for the underlying hardware. At the core of such hardware, we find a multitude of microchips, commonly referred to as digital Integrated Circuits (ICs). To provide assurance in hardware, analysts often employ Hardware Reverse Engineering (HRE), which is a crucial technique to detect, e. g., counterfeits [28], intellectual property violations [73], or even malicious circuit manipulations [58] in ICs. HRE consists of two fundamental steps [76]: First, analysts reconstruct a blueprint of the thousands to millions of logic gates on the IC – the so-called *netlist*. Second, they analyze this often very complex netlist using a variety of methods tailored to their detection objectives. Even though tools such as HAL [23, 80] exist, which partially assist human analysts during HRE, success depends heavily on their experience, skills, and cognitive abilities for performing complex and diverse HRE subprocesses [22, 86]. Thus, similar to other application areas such as design engineering and medicine, HRE represents a specific Human Computer Interaction (HCI) setting in which humans perform (computer-aided) complex problem solving using digital visualizations [24, 40].

When analysts navigate through a netlist, they have to integrate the current visual information with previously acquired knowledge about other parts of the netlist to make sense of its broader interconnections and functions. We argue that eye tracking is therefore highly suitable to gain a deeper understanding of how reverse engineers approach the analysis of a netlist. To interpret such eye tracking data holistically in terms of participants' strategies, reasoning, insights, or reflections on errors, more information is needed. A way to enrich eye tracking data with this kind of information is to let participants verbalize their thoughts as they solve an HRE task, or to ask them to subsequently reflect on the task. We propose that this combination of eye tracking and Think Aloud (TA) is a promising and comprehensive approach to cognitive-factors research in HRE. However, evidence from similar fields, e. g., design engineering, marketing and cognitive psychology, show conflicting results as to whether verbalizations influence the behavior of the problem solver during a given task [9, 15] or not [21, 60]. Therefore, it is not clear if and to what extent results from other studies can be transferred to HRE problem-solving processes. Hence, the primary objective of this work is to design, implement, and validate a mixed-methods approach that can subsequently be used to gain a comprehensive and nuanced understanding of the capabilities and skills influencing HRE success. Understanding how problem-solvers navigate in a netlist may inform the development of innovative hardware protection schemes. It may further help tailor educational programs [85] on hardware security.

To investigate if and in which way eye tracking, verbal TA protocols, and log files can be used to observe HRE problem solving, we conduct an experiment with 41 participants. Our main contributions are the following:

- We show that eye tracking provides high-resolution data on participants' visual attention to individual circuit elements.
- We identify Think Aloud as suitable for gaining insight into reverse engineers' thought processes, with eye gaze–cued Retrospective Think Aloud (RTA) generating a higher quantity of codes, while Concurrent Think Aloud (CTA) allows us to synchronize participants' verbalizations with their eye tracking data.
- We investigate the potential of analyzing eye tracking and TA data in combination and present two use cases: (1) mapping of TA codes with eye tracking for the triangulation of participants' navigation patterns, and (2) identifying strategies used in an HRE task with eye tracking, without needing to fully rely on TA.
- Drawing on previous research on TA and eye tracking, we advocate for methodological evaluations, propose enhanced combined analyses, and highlight the broader applicability of our approach to visually demanding problem-solving tasks in digital environments. In addition, our results provide insights for practitioners aiming at improving hardware security or developing educational tools pertinent to HRE.

2 BACKGROUND AND RELATED WORK

2.1 Hardware and Netlist Reverse Engineering

In its broadest sense, reverse engineering describes the process of recovering the underlying specifications and design process of any man-made object by anyone else than the original designers [59]. Hardware Reverse Engineering (HRE) in particular is often motivated by checking for the absence of malicious manipulations also known as hardware Trojans, which may have been introduced by a contracted manufacturer [58]. Further applications include the identification of Intellectual Property (IP) infringements such as counterfeit ICs which pose a major risk to the IC supply chain [28], or failure analysis of ICs aimed at improving manufacturing processes [11].

Azriel et al. describe the reverse engineering of digital ICs as a two-step process [3]. Initially, one obtains the circuit diagram of Boolean logic gates and memory elements and their interconnections – the gate-level netlist – from the IC. A single logic gate implements a small Boolean function such as NOT, AND, NAND or XOR. While single gates are straightforward to grasp for humans, understanding the complex functions that emerge when several are combined quickly becomes a major challenge. Netlists can be recovered using scanning electron microscopy of an actual IC [76] or, e. g., by gaining direct access to design files describing the netlist [3]. Reverse engineers then apply a wide range of methods for recovering and making sense of the higher-level structure and design rationale of the netlist at hand [1, 51, 73]. Given that this sense-making stage operates on the abstract digital description of

a circuit, rather than the physical sample, it is often referred to as *netlist* reverse engineering. Analysts need to rely strongly on their own experience, technical skills, and cognitive abilities for manual analyses, as well as to adequately set up and apply methods from their semi-automated toolbox [5, 22, 86].

State-of-the-art reverse engineering tools support analysts via interactive visualizations of the extracted circuit [63, 76, 80]. Those visualizations appear to lend themselves well to the graph-based structure of electronic circuits and make them more accessible [85].

2.2 Prior Research on Problem Solving in HRE

While the technical steps in HRE are often tedious to perform but generally well understood, little research has been done on the human problem-solving aspects. Lee and Johnson-Laird performed an early laboratory study investigating how HRE novices analyze simple Boolean circuits using fully manual approaches [43]. The authors define HRE as a specific and poorly understood kind of human problem solving, requiring analysts to identify how each component influences the output of the circuit at hand, as well as how the different components depend on each other.

Later research applied the findings of Lee and Johnson-Laird to more complex, real-world HRE settings [5, 84], providing first valuable insights into the higher-level strategies and cognitive processes. This strain of work led to a hierarchical model that divides the HRE process into *reversing actions,* such as inspection and information gathering or strategy decisions, and *source code development* [86]. A drawback of these initial investigations is that they are based on a complex training phase of human subjects and require massive evaluation efforts due to manual annotation of log files. This leads to small sample sizes and limits generalizability of these first studies on capabilities and skills in HRE.

Recent efforts addressing these challenges have resulted in the development of ReverSim [6], a game-based simulation that mimics realistic HRE subprocesses. Notably, ReverSim focuses on visually representing netlists, which are crucial for reverse engineers solving real-world HRE tasks. As ReverSim enables fine control over these visualizations, it facilitates the consistent collection of eye tracking data, which we consider a promising basis for studying important HRE subprocesses. At the same time, ReverSim allows recording and quantifying interactions of reverse engineers, regardless of prior knowledge, in a standardized environment.

2.3 Problem Solving and Eye Tracking

Eye tracking measures a person's visual focus non-intrusively. It estimates the positions of eye gazes from the captured eye images and the infrared reflections of pupils and corneas [48]. Eye tracking reveals rich and subtle dynamics of humans' cognitive processes when they review materials, e. g., on a computer screen, without causing discomfort or requiring particular effort from research participants. Thus, researchers have been developing and using eye tracking for decades to understand various psychological and physiological factors, including cognitive load [54], personality traits [8], health status [79], and problem solving [52, 88]. Recently, using eye tracking to study engineering and computer-assisted tasks has received growing attention, especially in software engineering [52]. For example, researchers adopted eye tracking to investigate the individual differences in comprehending software programs [77], which are related to different factors such as task familiarity [35] and age groups [55]. Sharafi et al. leveraged eye tracking to identify people's problem-solving strategies when they manipulate data structures for programming [67]. Further, prior work utilized eye-tracking to reveal how people debug software programs, relating their performance differences to the problem-solving strategies applied [47]. Beyond software engineering, previous research employed eye tracking in other tasks, particularly gamified tasks that involve significant visual interaction and navigation, to gain insights for educational, medical, and engineering applications [41, 44]. Recent studies have also proposed to use eye tracking in computer security research, including software reverse engineering [50].

The eye tracking metrics used for studying problem-solving processes are primarily related to visual attention and its transition. Fundamentally, eye tracking reports a time series of eye gaze positions. The three most common abstractions of this time series are fixations, saccades, and scanpaths [68–70]. Fixations are clusters of relatively stable eye gazes, standing for a basic unit of visual attention. Saccades are rapidly moving eye gazes in between two consecutive fixations. A scanpath is the resulting sequence of fixations due to the transition of visual attention. Multiple spatial and temporal metrics are computed on the top of these abstractions, e. g., number of fixations, duration of fixations, and attention switching [68–70]. Note that these metrics are often evaluated regarding the composition of visual stimuli, where researchers define their Areas of Interest (AOIs) [62]. It enables them to analyze the fixation and saccade metrics within each AOI or across different ones [68–70]. Fixation metrics are more commonly adopted than saccades, as they retain richer information of cognitive processing [33, 66]. Prior research motivated us to employ eye tracking, especially fixation metrics, for HRE problem solving – a task that is in particular visually demanding. However, those eye tracking metrics alone may lack interpretability as a primary method to understand problem solving, though offering detailed measurements in its spatial context [14]. As such, prior work proposed to combine eye tracking with other study methods, e. g., Concurrent Think Aloud (CTA) or Retrospective Think Aloud (RTA) which elicits problem solvers' thoughts, to attain interpretable and fine-grained measurements at the same time [27, 60].

2.4 Concurrent and Retrospective Think Aloud

When conducting TA in research studies, participants are asked to verbalize their thoughts to obtain information about their motivations, strategies, problems, or in general the "how" and "why" for a specific action. This method was first introduced in 1920 by Watson [81] with the goal of making thinking observable. Since then, TA methods have been developed and evaluated in many domains and contexts. In addition to CTA, where participants are asked to verbalize their thoughts while solving a given task, RTA, where participants are asked to verbalize the thoughts they had *after* completing the task, has also been receiving attention in the research community [19]. A common extension of RTA is cued retrospective reporting, where participants are shown recordings of their problem-solving session, complemented by their eye movements and mouse operations. Eye-gaze cueing has been shown to produce more comprehensive reports and improve participants'

ability to recall their thoughts, even though the gaze cue can be distracting for some participants [17]. Both methods have individual advantages and limitations, and neither have been applied in the field of HRE. Therefore, one goal of our work is to investigate both methods in conjunction with eye tracking during HRE processes.

For CTA, it is assumed that the verbalizations are mainly about actions and their outcomes [74], decision-making steps [39] or generally representations of short-term memory contents [19]. We therefore assume a direct and unaltered access to participants' problem-solving approaches. Also, immediate verbalizations are more synchronous to the eye movements than RTA, where TA is cued *by* eye movement. Of course, it is conceivable that CTA might distract the participant from their task or otherwise affect their performance [19]. Previous findings promote these assumptions at least in the field of design problem solving [15]. Also, empirical studies have presented evidence that CTA might skew eye tracking data [57]. However, a number of studies that did not find influences on eye tracking data or problem-solving behavior [21, 42, 60] challenge those reports. Due to the inconsistent evidence, an investigation of CTA's influence on the HRE process is part of the study's research questions.

Verbalizations generated by RTA stem from both long-term and short-term memory [19] and are assumed to contain more statements on participants' final choices [39]. Cued retrospective reporting elicits even more action, "how" and metacognitive information than non-cued RTA [78]. In addition, we can rule out the possibility that participants' performance during the task is influenced by RTA and their performance can serve as baseline for comparison with the CTA group. For these reasons the cued RTA method was chosen for the present study and its evaluation was included as part of the research questions.

2.5 Research Questions

From prior methodological work *outside* of Hardware Reverse Engineering (HRE), we derive that Think Aloud (TA) and eye tracking may be promising techniques for investigating problem-solving processes involved in netlist reverse engineering. Neither method has previously been used in the domain of HRE problem solving. Thus, it appears highly desirable to investigate their usefulness in studying netlist reverse engineering behavior. Given inconclusive findings in prior research, evaluating the potential interactions between eye tracking and TA methods is essential for establishing a methodologically sound and robust experimental setup. To this end, we answer the following research questions:

RQ1 Can fixations obtained from eye tracking be used to observe behaviors within HRE problem solving?

RQ2 How do Concurrent Think Aloud and Retrospective Think Aloud differ in revealing behaviors and approaches within HRE problem solving?

RQ3 Does Concurrent Think Aloud influence participants' performance, user experience, or eye movement?

Based on the methodological insight from the three research questions above, we explore and discuss a mixed-methods design combining TA and eye tracking. We highlight the utility of both CTA and RTA, with a particular focus on the *combined* analysis of verbal protocols and gaze behavior. Thus, our overarching research question is:

RQ4 How can eye tracking and Think Aloud complement each other in describing HRE problem solving?

3 METHODS

We conducted a lab study in which eye tracking was employed in combination with CTA or gaze-cued RTA while participants solved HRE problems. In the following sections, we provide details about the materials used, our participants, study procedures, as well as data collection and analysis.

3.1 HRE Task Materials

3.1.1 HRE Simulation. To administer the HRE tasks central to the present work, we used ReverSim, a computer game-based simulation of netlist reverse engineering problems [6]. The simulation consists of multiple *levels*, each representing one HRE task. Each task comprises a Boolean circuit diagram that participants need to reverse engineer in order to advance to the next level. Figure 2 shows the user interface with an example circuit. Specifically, in each task the participants need to reason about the functionality of the circuit in order to identify the binary input values that light all light bulbs while *not* triggering any danger signs. To enter their solution, participants can interact with the three switches to the left. Closing a switch powers the connected wire, corresponding to a binary input value of 1. Each gate within the circuit takes the binary values on its input wires, applies its Boolean function, and generates a corresponding value on the output wire. We illustrate this process using a straw man example in Figure 3. The effect of the chosen inputs is displayed by highlighting all powered wires once participants submit their solution. Should the solution be incorrect, participants can start another attempt and revise the switch positions. Participants can also annotate each circuit by using the mouse to draw onto the screen in three different colors.

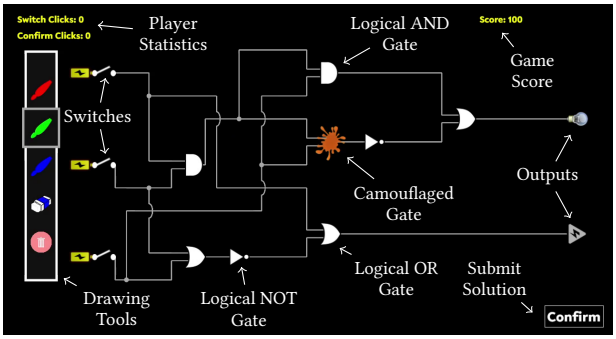


Figure 2: An example level of ReverSim. Participants need to understand the functionality of the circuit and then set the switches to the left such that the light bulb illuminates, whereas the danger sign must not be supplied with current. With the drawing tools they can annotate the circuit. The function of the gate in the form of an ink blot is hidden from the participants to make the solution of the level more difficult, simulating camouflaged gate obfuscation [13].

Figure 3: A straw man example of netlist reverse engineering. The goal is to light the bulb which needs a binary input value of 1. Knowing that the logical NOT gate can invert the signal from 0 to 1, we make the switch open (0) to light the bulb.

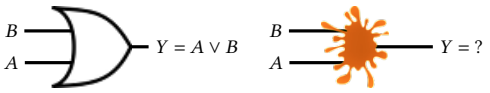


Figure 4: Two gate symbols from the circuits used in the HRE simulation. The left symbol depicts a standard logical OR gate with two inputs and a single output. The right symbol is specific to the ReverSim environment and represents a camouflaged gate [13]. The logic function of this circuit element is hidden from the participant, representing the case where the function of a gate could not be extracted from an IC due to an obfuscation countermeasures.

3.1.2 Netlist Reverse Engineering Tasks. The tasks used in this work were taken from the ReverSim level library. Each features three inputs and two outputs. First, we selected four medium-complexity tasks, where each possible combination of light bulbs and danger signs appears exactly once, i. e., there were no two tasks with identical target output values. Second, we included two tasks containing simulations of obfuscated circuit elements: Covert gates aim at confusing reverse engineers by mimicking one type of gate when visually inspected, while actually implementing a different functionality [65]. A camouflaged gate, on the other hand, is clearly identifiable as being obfuscated, but the actual functionality is hard to identify [13]. Figure 4 shows an example of a camouflaged gate as a game element in ReverSim, visualized as an ink blot.

3.2 Participants

In April and May 2022, we recruited 50 participants from a university in an English-speaking country. Conditions for participation were a minimum age of 18 and sufficient English proficiency. Each participant was compensated with 15 USD per hour. We aimed at recruiting a diverse population regarding prior knowledge. Therefore, we not only advertised the study in electrical and computer engineering courses but also in other departments and on campus via email, flyers, and word of mouth. The first nine participants were recruited to pilot the study. Hence, we obtained a sample of 41 participants for the analyses reported below. We randomly divided our participants into two groups with CTA and RTA, respectively. 20 participants were assigned to the CTA group and 21 were assigned to the RTA group.

After voluntarily agreeing on study participation, all participants answered a pre-study questionnaire on their age, gender and educational background including their majors of study. Table 1 shows a detailed breakdown of the demographics in our CTA and RTA groups. In general, our sample was young and educated. 28 of them had an educational background in electrical and computer engineering, while 13 had not. Furthermore, all participants self-rated their prior knowledge in 15 domains related to netlist reverse engineering. The prior-knowledge scale was developed alongside ReverSim and concerns areas such as Boolean algebra, digital circuits, as well as reverse engineering in general [6]. It yields a combined score as the mean of the individual answers on a five-point Likert scale ranging from "Very Low" (1) to "Very High" (5). All items offer a separate option "unfamiliar with the term", which is represented as 0. We report prior-knowledge distributions for both groups in Figure 5, showing "Medium" as the most common score.

Table 1: Basic demographics and prior knowledge score of our participants by assigned TA condition.

Condition	CTA		RTA	
	rel.	abs.	rel.	abs.
Number of Participants		20		21
Gender				
male	60%	12	57%	12
female	40%	8	43%	9
Age Range				
min		18		18
mean		24.7		23.2
max		32		34
Education				
secondary	40%	8	52%	11
tertiary	60%	12	48%	10
Prior Knowledge Score				
mean		3.15		3.24
SD		1.27		0.83

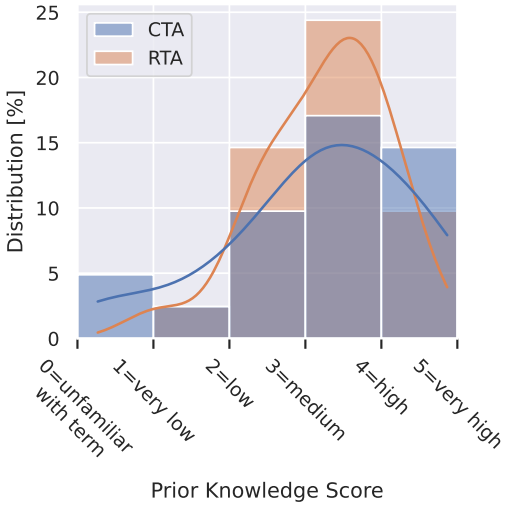


Figure 5: Distribution of prior-knowledge scores for both TA conditions. Most participants self-rated their prior knowledge between "medium" and "high".

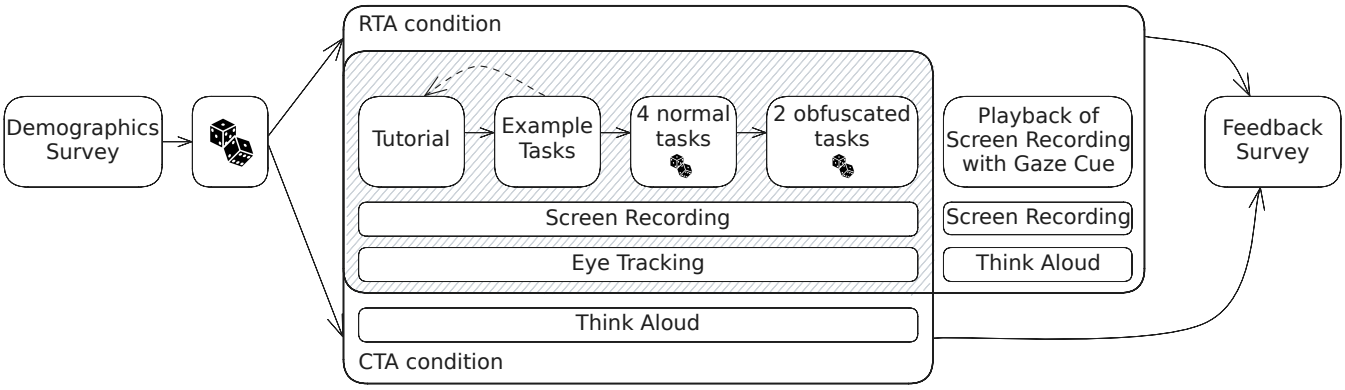


Figure 6: Overview of the study procedure. Participants first filled out a basic demographics and prior-knowledge survey and were then randomly assigned to the CTA or RTA condition. Both conditions contained the same set of tasks, during which screen captures and eye tracking data were recorded. Participants in the CTA condition thought aloud while performing the tasks. Those in the RTA condition thought aloud while reviewing a playback of their interactions with the tasks. Finally, all participants answered a feedback survey.

3.3 Ethical Considerations

Our study was approved by our institution's Institutional Review Board (IRB) and we secured participants' rights, safety and data privacy. Before participation, we informed participants about the study procedure, possible risks, and the right to withdraw at any time. We minimized participants' fatigue with regular rest periods throughout the study. We recorded the computer screen, surveys, eye-tracking data and audio during think-aloud sessions. Besides voice recordings, no personal information were collected. We transcribed the audio recordings using an automated transcription service and manually curated them. Through this process, we made sure no personal information is revealed in the transcripts for analysis and storage. Last, we stored all data on a secured department server.

3.4 Study Procedure

Accompanied by an experimenter, each participant completed the study procedure presented in Figure 6. Participants gave voluntary informed consent, followed by a survey on their background and demographics. Then, the experimenter walked participants through a tutorial about the basic concepts and interactions required to solve the tasks and gave instructions for thinking aloud. Importantly, participants did *not* receive explicit guidance on reverse engineering strategies. To check whether participants understood the basic concepts and interactions, the experimenter instructed them to pass three minimal working examples of netlist reverse engineering tasks; otherwise, participants revisited the tutorial. Participants in the CTA group were instructed to practice thinking aloud while solving these example tasks.

Next, we assigned the same six reverse engineering tasks described in Section 3.1.2 to both groups. We first presented the four tasks with medium complexity and then the two tasks with obfuscated gates, each in random order. The experimenter instructed participants to rest between two consecutive tasks. Participants in the CTA group thought aloud throughout the tasks. Conversely, participants in the RTA group worked in silence and were guided to think aloud after solving all tasks. When thinking aloud, these participants referred to a replay of their interactions with ReverSim with their eye movement overlaid on the screen recording to support them in recalling their behaviors. In accordance with the recommendations by van Gog et al. [78], participants were in control of the video playback. After solving the tasks and thinking aloud, participants filled out a post-study survey about their experience with the tasks and the TA method. Overall, each CTA experiment took around 1 to 1.5 hours and each RTA experiment consumed 1.5 to 2 hours.

3.5 Data Collection

During the experiment, we collected logs and screen captures within the ReverSim environment, as well as eye tracking and TA recordings, in the manner presented below. Furthermore, we collected participants' feedback after the study task. We took particular care to ensure that all data is accurately time-synchronized, such as to allow cross referencing between the various types of data.

3.5.1 Log Files. Throughout the experiment, ReverSim records all interactions, including drawing, interacting with switches, and submitting solutions. All such events are stored in a time-stamped *log file*, allowing us to precisely track participants' progress, solution time per level, and correctness of individual solutions.

3.5.2 Eye Tracking Data. We used a Tobii Pro Nano eye tracker to collect eye tracking data with a sampling frequency of 60 Hz. We mounted the eye tracker and displayed the stimuli from the HRE simulation on a 19-inch computer monitor with a resolution of 1280×1024 pixels. The monitor was placed about 50 cm away from participants' seating position. For each participant, we calibrated the eye tracker before collecting data, following a standard procedure [31, 32, 90]. We guided participants to take a seat and adjusted the eye tracker's tilt for them. Next, we calibrated the system using the Tobii Pro Eye Tracker Manager software [75]. The

software displays nine white dots on different parts of the screen as targets for the participant to stare at. It uses the calibration data to personalize the eye model for better tracking accuracy.

3.5.3 Think Aloud Protocols. To ensure TA quality, participants were asked to indicate their native language and to rate their proficiency in English. Based on their self-rating and our observations, all participants met general professional proficiency. We captured their TA verbalizations using a desk microphone and stored them in combination with a screen recording, such that the current state of the HRE simulation as well as any mouse movements were available to provide context for later analysis. Transcripts were automatically generated from the audio recordings for both CTA and RTA groups using Microsoft Office 365.

3.5.4 Feedback Survey. Following Ruckpaul et al. [60], we assessed whether participants were comfortable with both CTA and RTA settings. All participants gave feedback on their personal experience of the experiment in three areas using an online questionnaire: (1) satisfaction with their personal task performance, (2) perceived difficulty of the task, (3) confidence when describing the tasks during TA, (4) CTA specific: how helpful it was to think aloud to solve the tasks, and (5) RTA specific: how helpful the eye-gaze cued video playback was in remembering and describing what they were thinking. All items across all areas consisted of a rating on a five-point Likert scale and an optional free-form answer field.

3.6 Data Analysis

Below, we provide an overview of how we analyzed eye tracking and TA data, respectively and jointly.

3.6.1 Analysis of Eye Tracking Data. To answer **RQ1**, our analysis of eye tracking data follows the steps below.

Data synchronization and cleaning. Before analyzing our data, we temporally aligned the data sources using timestamps attached to each recording. We first synchronized the raw eye gaze data with the simulation log files recorded by ReverSim. This allowed us to subsequently extract the eye-tracking data for each HRE task. We removed data from individual HRE tasks for a small number of participants due to poor quality or interruptions during the task. For this purpose, we carefully cross-examined the logs from the eye tracker and the simulation server, screen recordings, and notes taken by the experimenter, resulting in the exclusion of 6 out of the 246 recorded tasks.

Fixation detection. To model participants' visual attention, we extracted their fixations from the raw eye gaze data. Our pipeline adopts the `I2MC`[1] algorithm and applies Kalman filtering, interpolation of missing data, and 2-means clustering to detect fixations and address noise from the eye tracking data for our analysis [29].

Defining Areas of Interest (AOIs). AOI analysis enables us to evaluate fixation metrics with regard to individual visual elements of the HRE simulation. We defined AOIs on the screen for each of the six HRE tasks. Our AOIs were grouped into three categories: logic gate elements, circuit input/output, and User Interface (UI) elements such as the controls for the drawing tools. Figure 7 showcases the AOIs for one task. Each AOI covers a rectangular area around the center of each element. We follow Goldberg and Helfman's guidelines to determine the granularity and sizes of AOIs [26], despite there being no universal practice for it.

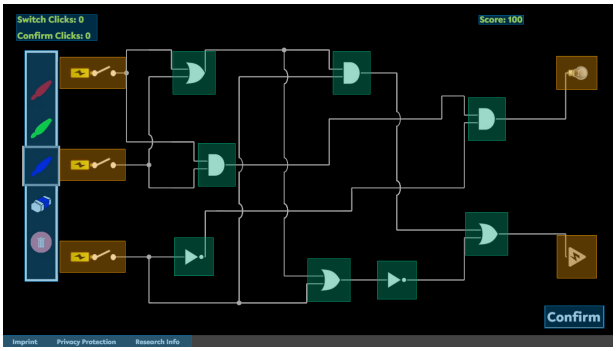


Figure 7: Areas of Interest (AOIs) defined in one of the HRE tasks. The color indicates the AOI category: UI elements (blue), inputs and outputs (orange), and logic gates (green).

Fixation metrics and analysis. We leveraged fixation metrics to evaluate participants' visual interest on AOIs. We first measured *fixation rate*, which is the ratio of the total number of fixations in each AOI to all AOIs of circuit elements [68]. A higher fixation rate suggests participants' greater visual interest in an AOI. Similarly, we computed the *proportional fixation time*, which is the ratio of the sum of fixation durations in one AOI to the total duration time in all AOIs of circuit elements [7]. In addition, we generated fixation heatmaps on images of the HRE tasks to aid visual analysis. In **RQ1**, we examine (1) how HRE tasks with different logic gate complexities affect participants' eye tracking metrics and, and (2) the correlation of eye tracking features between different gates in relation to participants' problem-solving processes..

3.6.2 Iterative Coding of Think Aloud Protocols. We analyzed the TA transcripts by qualitative coding and content analysis [38] to inductively develop a codebook in the light of **RQ2**. One researcher checked and corrected the automatic transcriptions from both TA data sets for errors. The most common error sources were technical terms such as *AND gate*, *inverter* or *wires*. Simultaneously, the researcher applied content-based segmentation in order to prepare the transcripts for coding. From the six recorded tasks for each participant, we selected both the first task they encountered and the task containing obfuscation (a camouflaged gate). This way we wanted to capture the problem-solving behavior of the participants when they encountered a new task of each set for the first time.

Four coders individually applied an open coding procedure on those transcripts to categorize verbalizations. We used screen recordings alongside to resolve verbal references to individual on-screen components. All coders discussed the elicited codes and resolved disagreements. To ensure that we accurately capture differences between CTA and RTA, the four coders generated a coding scheme for each separate TA method before merging both codebooks.

[1] `I2MC` is available under open-source license: https://github.com/royhessels/I2MC/

Two coders individually applied the coding scheme on a training sample of 20% of the transcripts and iteratively discussed differences in their coding in order to revise the codebook. After this first phase Inter-Rater Reliability (IRR) reached $\kappa = .43$ (Cohen's kappa). We further refined the codebook to improve reliability by merging similar codes, further discussing disagreement and resolving ambiguity in code definitions. Subsequent to this second phase, the codebook was applied on 20% of the sample to validate reliability. Overall IRR improved to $\kappa = .51$. However, specific codes concerning participants' navigation behavior achieved an IRR of $\kappa >= .71$. Given that those codes are essential for mapping with the eye tracking data, we decided to apply the codebook with one coder and limited our analysis to those codes. After creation of the final codebook (see Appendix A), one of the involved coders deductively coded 48 tasks from 24 randomly chosen participants – 12 from the CTA and 12 from the RTA group. In our analysis and discussion, we report the frequencies of codes and compare the results from both TA methods in relation to prior literature.

3.6.3 Effects of the CTA Method on Participants and Data. As highlighted in Section 2.3, asking participants to think aloud while solving a cognitively demanding task might affect their performance or impact eye tracking data. In the light of **RQ3**, we therefore analyzed the effect of CTA on our participants' experiences, as well as on their task performance and eye tracking data. To verify that we did not create a negative experience for participants by asking them to concurrently think aloud while solving the HRE challenges, we analyzed their answers to the feedback survey. In particular, we compared mean values of participants' self-reported confidence regarding the use of both TA methods, as well as the perceived difficulty of the HRE tasks.

Exercising caution regarding potentially detrimental effects of CTA on the experiment itself, we then checked for differences between data acquired from the CTA group compared to the RTA group. Here, the latter group serves as our non-CTA control, given that those participants worked in silence. Regarding task performance, we evaluated CTA's impact on participants' problem-solving time. Furthermore, we counted the number of clicks on switches *over par*, i. e., the clicks that were unnecessary for an optimal solution, as well as the number of attempts required to solve each individual task and applied the same comparison. In both cases, we used t tests to determine whether the sample means differ significantly between the CTA and RTA groups.

Following the approach by Ruckpaul et al. [60], we further verified whether CTA prolongs fixation duration compared to working in silence and thereby affects eye-tracking data. For this purpose, we calculated the relative frequencies of fixation durations captured in either TA conditions and compared the respective distributions.

3.6.4 Use Cases for Combining Eye Tracking and TA. To illustrate the potential of joint analysis of eye tracking and TA data in the light of **RQ4**, we present two use cases studying how eye tracking features are correlated with data from TA coding.

Our first use case concerns the level of individual code assignments. Here we examined how fixation statistics, e. g., their position distributions, correspond to participants' behaviors observed in TA. This approach relies on accurate time synchronization between eye gaze recordings and thought protocols. While synchronicity can be manually achieved for RTA, we focused our analysis on CTA transcripts, where automatic synchronization is possible. We extracted the start and end time for each occurrence of the relevant codes in the TA transcript using the timestamps inserted by the automatic transcription service. We then mapped the corresponding fragments of eye tracking data to each code occurrence and compared the fixation statistics between different codes.

In our second, high-level use case, we compared eye tracking data from participants who started an HRE task with different strategies. We first identified the initially applied strategy from the beginnings of RTA and CTA transcripts, using screen recordings for context where required. After grouping participants by strategy, we then applied AOI analysis to determine mean fixation durations for different AOIs and compared them across the groups.

3.6.5 Statistical Calculations. For statistical group comparisons, we used t tests when normal distribution and homogeneity of variance were given. We tested these with the Lilliefors test [46] and Levene's test [45]. If the assumption of normal distribution was violated we used Welch's t tests [82]; if further homogeneity of variance was not given, Mann-Whitney U tests [49] were employed. When multiple tests were calculated, Holm-Bonferroni [30] correction was applied. If variables were not metric, e. g., the code distribution between CTA and RTA, we used Pearson's χ^2 tests for comparison. For the eye tracking data we also calculated Pearson's r [56]. In general, we applied statistical calculations sparingly because (1) our sample was not very large, which is a risk for beta errors, and (2) we wanted to prevent alpha error accumulation due to multiple testing.

3.7 Limitations

We recognize several limitations in our study. First, our participant population is overall well-educated and young. Participants from other age groups or with other prior knowledge backgrounds may contribute to different observations in completing the HRE tasks. Second, though all participants met our requirements for English proficiency, it remains a question if speaking a non-native language affected their accuracy of TA in this context. Also, it remains unclear whether verbalizations are representations of unbiased retrievals from memory during TA. Nevertheless, future studies may use eye tracking to cross-validate the verbalizations during TA. Third, our work primarily analyzed participants' visual attention during HRE problem-solving using fixations captured with eye tracking and AOI analysis in line with prior work and guidelines [52, 70]. We chose this particular focus because these features are most crucial to understanding visual attention, which is the foundation for solving HRE tasks, and how to analyze them in combination with TA was previously unknown. Compared to fixations, the amount of information acquisition and cognitive processing during saccades is limited [33, 66]. Nevertheless, we encourage future studies to enable more kinds of analysis for HRE from aspects other than attention, including stress and cognitive familiarity indicated by saccadic and pupil features respectively [34, 71]. In addition, HRE problem solving may depend on other factors as well, such as the prior experience and working memory of the analyst [5]. Quantifying the influences of these factors is not within the scope of this paper. Despite these limitations due to the exploratory nature of our work,

we believe our study's contribution is significant as the first work to design mixed methods for understanding HRE problem-solving.

4 RESULTS AND DISCUSSION

4.1 RQ1: Analyzing HRE Problem Solving using Eye Tracking

This section discusses how to interpret participants' processes to solve HRE tasks from eye tracking features of individual circuit elements.

In Figure 8, we present one participant's fixation heatmap as an example for visualizing their attention on different regions of an HRE task. This particular task features a camouflaged gate, which means that its actual logic function is obscured by an ink blot. The example shows that the participant spent most fixations on the camouflaged gate, compared to other gates. An insight gained from this is that even though uncovering the logic function of the camouflaged gate is not actually required to solve the HRE task, the gate nevertheless presents a strong distractor.

While heatmaps are a helpful tool to visualize individual participants' behavior, statistical analyses on eye gaze behavior require an abstraction of AOIs: In the following, we compute the number and duration of fixations on individual AOIs. From this, we gain insight into the effect of gate types and obfuscation on eye gaze in Section 4.1.1. In Section 4.1.2, we investigate the effect of gate positions and their interconnection.

4.1.1 Complexity of Gate Types. We first demonstrate how tasks with and without obfuscated gates affect participants' visual attention. Figure 9 shows the statistics of fixation rates for each category of AOIs for the three different task types; i. e., without obfuscation, camouflaging obfuscation and covert obfuscation. Overall, we observed that the fixation rates feature similar proportions between the AOIs in all three types of task. A comparison of proportional fixation times, shown in Appendix B, yields equal results. For the tasks containing either type of obfuscated gate, those gates occupied a major fraction of visual attention compared to non-obfuscated ones. This matches the initial observation from Figure 8 where we concluded from the heat map that a considerable amount of attention is drawn to the camouflaged gate.

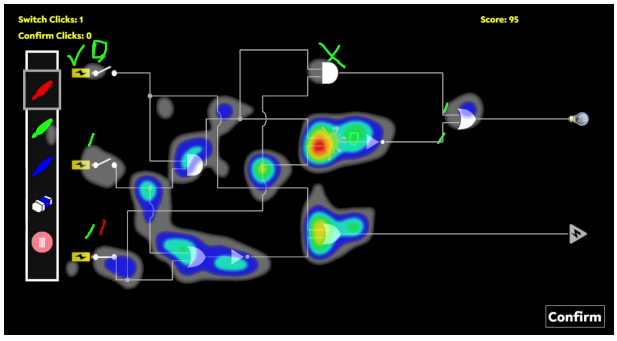


Figure 8: Heat map of raw eye gaze data for a single participant in an HRE task involving a camouflaged gate, overlaid with the participant's annotations on the circuit. The camouflaged gate is represented by the orange ink blot in the center. It is evident that this participant focuses most of their attention on the camouflaged gate, while the input and output elements receive very limited attention.

In addition, participants spent the least visual attention on UI elements, despite the total size of this category being the largest. Also, participants were more attentive to the input switches than the circuit's outputs. This stands to reason because participants need to enter their solutions by clicking switches, while the outputs are static elements that require little reasoning. Note that the variances are large in every type of task, revealing substantial individual differences between participants.

4.1.2 Correlation Between Gates. Beyond showing the influence of individual elements on problem solving, we employed eye tracking to investigate the underlying association between circuit elements. We computed the correlations of fixation rate and proportional fixation time between every pair of circuit element AOI for all levels. Specifically, Figure 10 shows a correlation coefficient matrix for fixation rate in one task, across participants. We find that the significant correlations visible in Figure 10a correspond to gate pairs between which exists an immediate, or at least short, connection.

4.1.3 Discussion of RQ1. From the above results, we summarize the following takeaways. First, both heatmaps and AOI analysis helped us identify where participants spent most visual attention during HRE tasks. Notably, both analyses indicate that eye tracking on circuit diagrams as employed in the present setting is precise enough to resolve participants' focus on individual circuit elements. From a problem-solving perspective, the distractive power of individual obfuscated gates is remarkable. Even more so, camouflaged gates appear to draw participants' attention regardless of whether an understanding of their function is in fact required to solve the HRE task at hand. This insight may give rise to more efficient hardware protection schemes, which we will further discuss towards the end of this paper.

Second, we discovered correlations between fixation rates on different gates. As these correlations correspond well with the interconnections of the gates within the netlist, we see this as a further indicator that eye tracking may be suitable to track participants' navigation within a netlist, for which we provide two use cases when answering **RQ4**. While in the present example, we performed the required AOI analyses after completion of the experiment, we consider it certainly feasible to do so in real time with dynamic AOIs. The ability to perform such fast component-level AOI analyses may, furthermore, open up a new avenue for integrating eye tracking into reverse engineering tools such as HAL [23].

4.2 RQ2: Revealing Behaviors and Approaches through Participants' Think Aloud

To further reveal participants' behaviors and approaches during HRE problem solving, we analyzed their TA transcripts using a coding technique based on qualitative content analysis as described in Section 3.6.2.

4.2.1 Final Codebook and Code Frequencies. The final codebook consists of the 16 main codes shown in Figure 11 with their observed relative frequencies. Four of these codes were specific to the

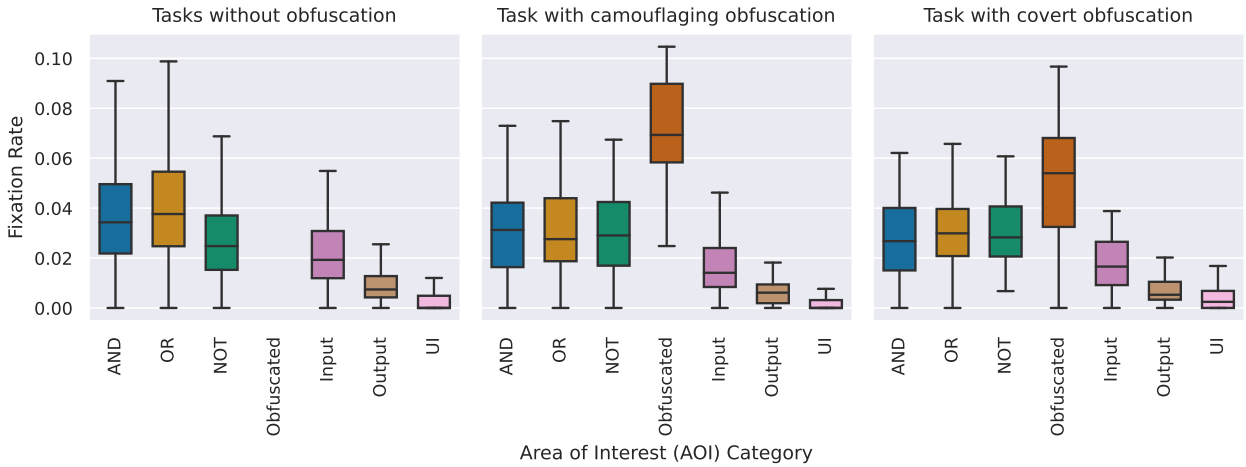


Figure 9: Statistics of fixation rate for each AOI category under different task complexities (RTA group). The basic logic gate types (AND, OR, NOT) receive similar attention across all types of the HRE tasks, but are outweighed by both camouflaged and covert gates. Output and UI elements receive little attention. In the statistics for proportional fixation time and the CTA group, we attained very similar observations to this example.

Retrospective Think Aloud method. The full codebook and code hierarchy including all sub-codes can be found in Appendix A.

For forward tracking we reached Cohen's kappa of $\kappa = .71$ and for backward tracking $\kappa = .73$. These codes were assigned to capture participants' navigation within the task, which is fundamental for understanding problem-solving processes in HRE. In particular, forward tracking was assigned when participants proceeded with a given input value from a switch or gate and tracked this signal further towards the next gate or the circuit's output; i. e., light bulb or danger sign. Backward tracking was assigned when participants traced from the circuit's output or a gate towards the preceding gate or the switches.

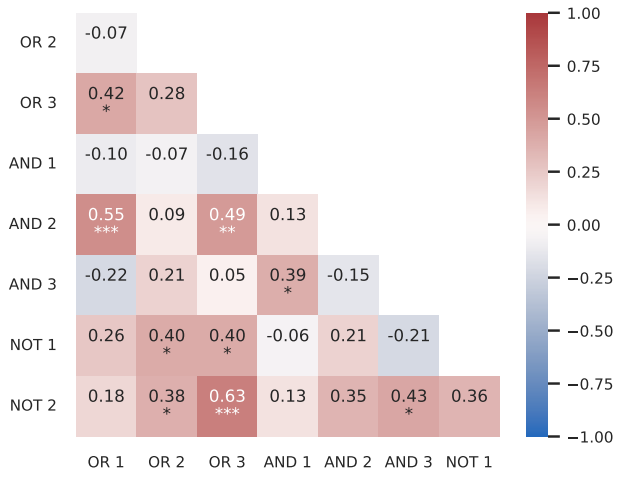
We compared the relative proportions between the codes common to both CTA and RTA and found no significant differences ($\chi^2 = 15.01, df = 15, p = .45$). Interestingly, CTA produces an average of 38.7 codes while RTA yields 45.9, which equates to approximately 19% more codes in the RTA condition. Note that 11.5 percentage points of additional codes are allotted to the RTA-specific codes, which describe behavior that occurred as participants recalled the task. The code *insight*, shown at the bottom of Figure 11, is by far the most prevalent of those codes and marks segments where participants came to new insights about the task or the quality of their solution while watching the video playback of their own actions. This difference is largely explained by RTA transcripts being significantly longer because participants were able to pause the video at will. In contrast, the length of CTA transcripts is limited by the duration of the HRE tasks. RTA participants used the pause feature a total of 118 times, corresponding to 5-6 times per participant across all coded tasks.

We further found that rate of speech, measured in words per minute, differed only slightly between CTA (94) and RTA (98). However, the presence of covert gates appears to influence rate of speech within the CTA condition. During the task with the covert gate participants verbalized approximately 24 words less per minute ($m = 72.4, sd = 30.5$) than in tasks without obfuscation ($m = 96, sd = 33.8$) or the task containing a camouflaged gate ($m = 96.1, sd = 37.4$).

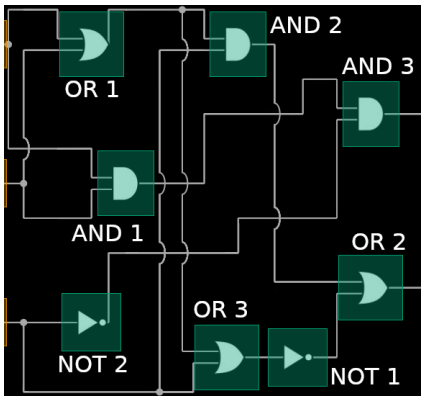
4.2.2 Discussion of RQ2. The overall high complexity of HRE tasks and participants' diverse strategies for solving them are reflected in our codebook. Thus, for some codes, it is natural to allow ambiguity when coders are required to interpret participants' mental activities from TA. We observed from our iterative coding process that resolving this ambiguity is possible via extensive discussion of disagreements with at least two coders, however, doing so incurs a significant overhead. In addition, the quality of codes applied to the RTA transcripts may differ from those in CTA due to some erroneous recollections during RTA (Section 4.4.3). Nevertheless, our current coding still reliably identified fundamental behaviors of HRE, namely forward and backward tracking, from the TA verbalizations. When applying our codebook, coders should make a trade-off between saving efforts by reducing the codebook complexity and gaining more in-depth insights from intensive discussion. In addition, both CTA and RTA appear to be appropriate TA methods to study HRE problem-solving, considering the following trade-offs. CTA offers immediate verbalizations of participants' navigation and reasoning and allows high temporal synchronicity with eye tracking. Nevertheless, in very complex tasks, RTA may offer more benefits than CTA, as CTA participants might stop talking due to their strong focus on the task.

4.3 RQ3: Differences Between CTA and RTA Groups

To determine whether CTA had an impact on participants' HRE, we compared the CTA and RTA groups in terms of their performance, user experience, and eye-tracking data. As the RTA group did not



(a) Correlation matrix. Asterisks in the matrix indicate significance levels: *:$p < 0.1$; **:$p < 0.05$; ***:$p < 0.01$



(b) Condensed version of Figure 7 showing the corresponding AOI in the underlying task.

Figure 10: Correlation matrix between the fixation rates on all gate AOIs within one task. Significant correlations between fixation rates on AOIs are explained by their underlying gates being direct successors or having common inputs.

think aloud while solving the levels but afterwards, their HRE behavior served as a baseline for our comparison.

4.3.1 Performance. To measure performance of solving each of the six levels, we resort to three metrics: time, attempts, and switch clicks over par (see Section 3.6). Figure 12 shows boxplots of the solution times per level for both groups. To determine whether there were significant differences between the conditions, we calculated multiple Mann-Whitney-U-tests [49] with Holm-Bonferroni correction [30]. Even without correction, the groups did not differ significantly in their performance regarding time (ranges: $U = [201.0 - 259.0], p = [.10 - .56]$).

Similarly, for switch clicks over par (ranges: $U = [126.0 - 205.0], p = [.16 - .85]$) and for number of attempts (ranges:

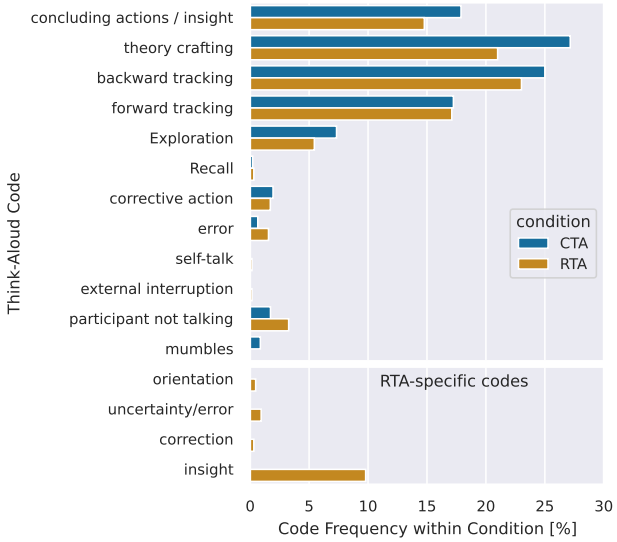


Figure 11: Relative frequencies for each think-aloud code assignment compared between CTA and RTA conditions. Bar lengths indicate relative frequencies. The bottom four codes are exclusive to the RTA codebook. Either condition generates similar relative frequencies across all codes, even though, in absolute terms, RTA generates more code assignments overall.

$U = [138.5 - 192.5], p = [.28 - .68]$) we observed no significant differences between CTA and RTA groups.

4.3.2 Participants' Feedback on Task Difficulty. In the feedback survey, we asked participants to rate the difficulty of the different task sets on a 5-point Likert scale ranging from 1 (strongly disagree) and 5 (strongly agree). As the performance of CTA and RTA participants did not differ significantly, we did not calculate inferential statistics for perceived task difficulty, but instead report mean values and standard deviations to reflect the participants' evaluations.

For the question, "Solving the first four tasks (puzzles) (...) was challenging to me.", participants from the CTA group ($m = 2.45, sd = 1.28$) as well as from the RTA group ($m = 2.43, sd = 1.25$) rather tended to disagree with mean answers between "neither agree nor disagree" and "somewhat disagree."

Conversely, for the question "Solving the last two tasks (puzzles) (...) [with an obfuscated gate] was challenging to me.", participants from the CTA group ($m = 3.50, sd = 1.15$) as well as from the RTA group ($m = 3.86, sd = 0.91$) rather tended to agree with means between "neither agree nor disagree" and "somewhat agree."

4.3.3 Participants' Feedback on the TA Procedure. We also report means and standard deviations of participant's self-rated confidence in describing the tasks by thinking aloud and participant's self-rated ease of verbalization for both groups, respectively.

For the question "I felt confident when describing the tasks (puzzles) during think aloud.", participants from the CTA group ($m = 3.95, sd = 1.15$) as well as from the RTA group ($m = 3.71, sd = 1.23$) tended to "somewhat agree."

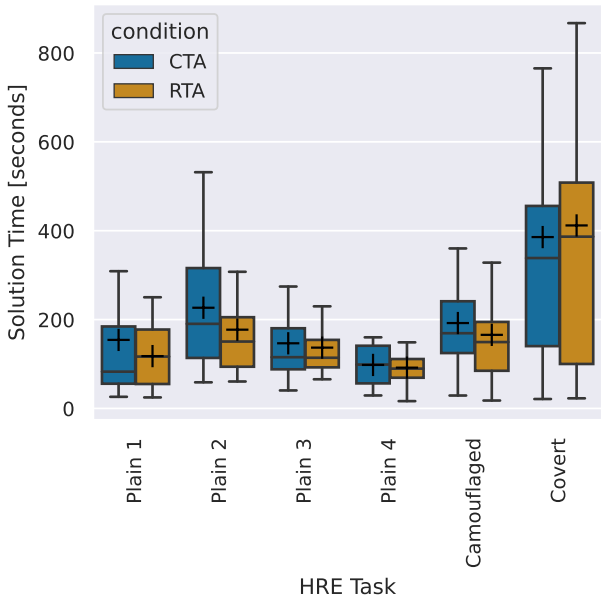


Figure 12: Distribution of participants' solution times for each HRE task, divided by TA condition. The four "Plain" tasks on the left do not feature obfuscation. An example task is given in Figure 7. The two tasks on the right feature a single obfuscated gate. An example for the camouflaged gate task is given in Figure 2. We observe similar solution times for both groups.

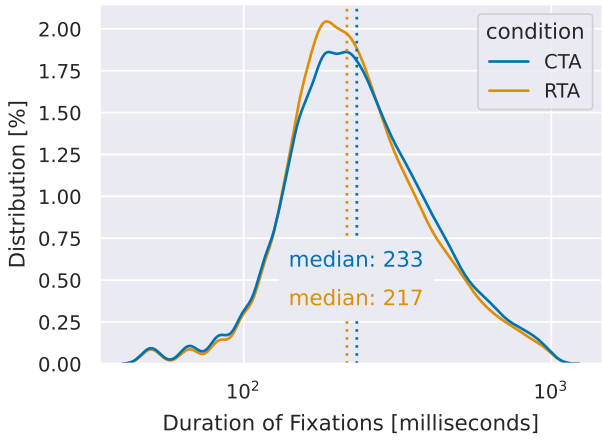


Figure 13: Distribution of fixation duration (kernel density estimation) for all participants in the CTA and RTA conditions (logarithmic scale). The two distributions are similar and approximately lognormal. The median fixation durations differ by about 16 milliseconds.

For the question, "I found it easy to verbalize my thoughts.", participants from the CTA group ($m = 3.75, sd = 1.33$) as well as from the RTA group ($m = 3.57, sd = 1.29$) also tended to "somewhat agree."

Although the mean values of the Likert scale responses are very similar, the content of the responses differ between groups, when asked for explanation of their rating. A participant in the CTA group who "strongly agreed" regarding their TA confidence argued *"I felt more like I was explaining the logic aspects of the puzzle in my thoughts rather than random solutions so I felt comfortable with that, with an exception of the last task."* This trade-off between difficulty and confidence was also mentioned by a participant who "somewhat disagreed": *"I feel like I was too focused on trying to figure the puzzle out in my head than having to say it out loud. I got better at speaking when I knew what I understand. But when I get something wrong I would be confused and not know what to say."*

Participants in the RTA group expressed problems remembering their thoughts during the task; e. g., *"At some points I wasn't entirely sure what I was thinking," "(...)if this had been done during my solving of the problems, it would have gone much better (...)"*

The CTA group was further asked to rate the statement "I found it helpful to think aloud for solving the task (puzzle)." With a mean of $m = 3.5, sd = 1.43$ they scale between "neither agree nor disagre" and "rather agree." One participant who "strongly agreed" mentioned: *"Yes, thinking aloud and writing small notes helped me keep track of my strategy."* In contrast, for a participant who "strongly disagreed" it felt *"(...) hard doing verbalizing and thinking at same time."*

The RTA group was asked to rate the statement "I found it helpful to refer to the video playback with eye gaze cues when remembering and describing what I thought." With a mean of $m = 3.9, sd = 1.22$ they "rather agree." Nevertheless, participants gave diverging explanations to this question in the free-form answer field. One the one hand they stated that it was *"definitely helpful because it reminded me of the way I approached the problem and what I started looking at and solving."* but on the other hand *"(...) that it was more distracting than helpful."*

4.3.4 Eye Tracking Data. To identify whether the TA method systematically influences eye gaze behavior, we compared relative frequency distributions of all fixations recorded within the CTA and RTA groups. We find that both frequencies approximately follow a lognormal distribution as shown in Figure 13 and observe very similar shapes (CTA: *mean* = 274ms, *median* = 233ms, *sd* = 158ms; RTA: *mean* = 267ms, *median* = 217ms, *sd* = 152ms). A Mann-Whitney U test shows that the observed 9-millisecond difference of mean values, with CTA exhibiting higher fixation lengths, is indeed strongly significant ($U = 5.7 \cdot 10^8, p < 10^{-7}$). This significance is probably due to the large sample of $n = 66,806$ individual fixations and should be interpreted with caution. The median fixation duration shows a similar result with a difference of about 16 milliseconds.

4.3.5 Discussion of RQ3. From the above analysis we identify little impact of the CTA method for all four tested effects. Participants' performance in both the obfuscated and non-obfuscated tasks did not differ significantly between CTA and working in silence, which is also reflected in their perceived task difficulty.

Regarding the effect on eye tracking, Ruckpaul et al. observed mean CTA fixations in their task that were about 55 milliseconds *shorter* than in RTA, however, did not obtain a statistically significant result ($p = 0.123$) [60]. Contrary to those findings, our results indicate a slight positive and statistically significant difference in means. We argue that, for the purpose of observing how visual attention is distributed on individual circuit components, this difference is of limited importance and comparability between CTA and RTA eye tracking data is generally given. In summary, we have no evidence that CTA systematically skews the data which our mixed-methods approach captures.

4.4 RQ4: Eye Tracking and TA as Complementary Research Methods

By addressing RQ 1, 2 and 3 we showed that eye tracking and TA are in isolation appropriate methods to investigate human problem solving in HRE. Combining eye tracking and TA data might therefore be useful for achieving a more holistic explanation of HRE behavior. To answer the overarching research question on how the strengths of each method can be complemented, we present two use cases. In Section 4.4.1, we perform a descriptive analysis localizing the prevalence of behaviors observed in different parts of the circuit, combining positional data obtained from eye tracking with time frames of individual behaviors observed in TA. In Section 4.4.2, we show that, on a higher level of abstraction, participants' chosen starting points in the HRE task leave characteristic patterns in eye tracking data. Additionally, in Section 4.4.3 we highlight how eye tracking can support RTA in the form of eye gaze–cueing.

4.4.1 Use Case I: Matching TA Codes to Eye Tracking Fixations. An advantage of accurate time synchronization between eye tracking and CTA protocols is the ability to precisely extract the eye tracking data for individual behaviors that were coded in the TA protocols.

Setup. In Section 4.2, we have identified the codes *forward tracking* and *backward tracking* as frequent behaviors that we can detect in the TA transcripts with high agreement. However, localizing those behaviors is extraordinarily tedious by manual coding of transcripts. To demonstrate how eye tracking reflects this information, we followed the steps below. We first obtained two sets of fixation time series labeled *forward tracking* and *backward tracking* using the timestamped code assignments from all TA transcripts. Each fixation represents visual attention to a specific coordinate on the screen. For each of the two sets, we then calculated the distributions of the fixations along the horizontal axis of the screen. By comparing the resulting density plots, we investigate the assumption that *forward tracking* occurs more often towards the inputs of the circuits, i. e., to the left, and that *backward tracking* is prevalent towards the outputs on the right.

Results. Figure 14 shows which locations on the circuit participants tended to fixate during episodes of forward or backward tracking. From the plots it is apparent that *forward tracking* is more prevalent at lower horizontal coordinates, i. e., towards the inputs of the circuit, whereas *backward tracking* is more often occurring

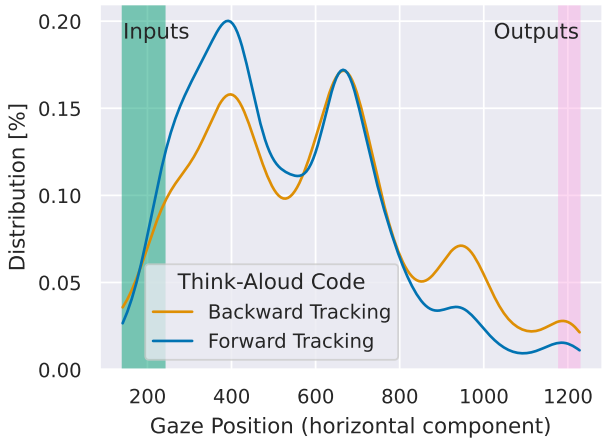


Figure 14: Distribution of the horizontal coordinate of fixations coded with forward and backward tracking actions (kernel density estimation), combined across all participants and all coded HRE tasks. The shaded areas on the left and right show the position of the circuit's inputs, i. e.switches, and the output symbols. During forward tracking, participants' visual focus is predominantly on the left side of the screen, towards the circuit's inputs and first set of gates. Backward tracking is more prevalent towards the right side of the screen, containing gates directly connected to either of the outputs. The three distinct peaks in both distributions are caused by the most prevalent horizontal position of gates across the different tasks.

towards the outputs of the circuit.[2] This result reflects our definition of forward and backward tracking behavior well. We believe that with two coders and consensus after discussion, an even better fit of eye tracking and TA could be reached. Eye tracking data could further serve as an indicator for coding quality if only one coder is available.

4.4.2 Use Case II: Identifying Strategies. To find out whether eye tracking is suitable for identifying task-specific strategies in HRE problem solving, we explored individual participants' gaze behavior.

Setup. First, we identified a sample task with a peculiar structure and formulated hypotheses for potential solution strategies: It is sufficient to reverse engineer the blue branch in Figure 15 to unambiguously solve the task, i. e., all switch positions can be clearly determined by reverse engineering this one branch. Conversely, the branch that ends in a danger sign alone does not provide sufficient information to solve the task. Arguably, the best strategy to solve this level is to apply *backward tracking* to the blue branch from the light bulb, while the other branch can be ignored. Second, we grouped participants by strategy, using the TA transcripts to identify at which of the two branches they started reverse engineering.

[2]Please note that each density plot is normalized, such that the area under both curves is 1. This corrects for the fact that in total participants spent more time *backward tracking* than *forward tracking*, as Figure 11 suggests.

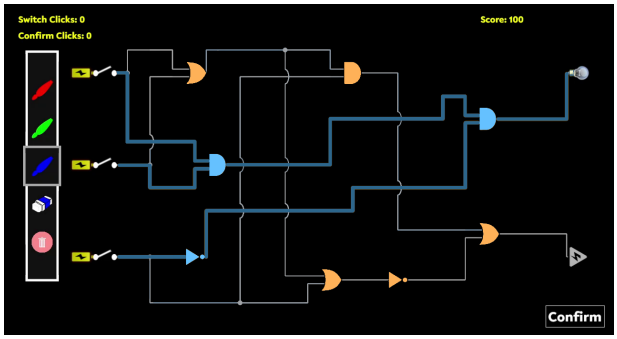


Figure 15: The structure of this HRE task allows a shortcut: All three switch positions can be unambiguously determined by following the blue path backwards from the light bulb on the right. Inspecting the remaining gates colored orange is not required and does not yield additional information.

If participants did not verbalize where they started, we extracted this information from the video.

Third, we used the eye tracking data to calculate fixation duration within the first 20 seconds during the task on the AOIs on both branches to capture how participants initially explore the circuit. We then compared the average fixation duration of participants who started at the light bulb versus participants who started at the danger sign.

Results. We found that participants who started reverse engineering at the light bulb spent an average of 1.44 seconds on each AOI on the path behind the bulb, while they fixated the other AOIs for 0.52 seconds each. Conversely, participants starting at the danger sign, spent an average of 0.99 seconds on each AOI on the light bulb path and 0.65 seconds on each AOI on the path feeding the danger sign.

In summary, the two strategies yield distinctive eye gaze patterns: Participants starting with the light bulb gazed more at the components of the blue branch in Figure 15, whether consciously or unconsciously. Participants starting with the danger sign gazed less at the components of the blue branch and more at the components of lesser importance. We argue that one may extend this approach to generate eye gaze models for different HRE problem-solving strategies identified from TA within a small number of participants. Using such models, automated analyses of eye tracking data could then enable the discovery of corresponding strategies within a large sample, where manual coding of full TA protocols for the same purpose would be prohibitively time-consuming.

4.4.3 The Case for Gaze-Cued RTA. During RTA, participants sometimes had difficulty remembering or verbalizing their problem-solving approaches during the task. We coded these RTA specific verbalizations as RTA orientation, RTA uncertainty/error, RTA correction, or RTA insight (see Section 4.2.1). This way, we could identify several instances in which participants reconsidered what they had just said because the captured eye tracking differed from their memories. To give an example from a participant who watched their video starting the first level: *"(...) and I also tried to pay attention close to the beginning – but I guess based on where my eyes were, that's not true – where or what the end points were like."* The participant remembered that they first had to identify the goal state of the task, i. e. light bulb or danger sign, to subsequently start reverse engineering from there. However, as this was their first task, they likely had not yet developed this strategy and only used it in subsequent tasks. During RTA, the participant became aware of this mismatch of their memory and their actual behavior visible from their eye gaze recording. This example indicates the potential improvement of RTA by eye tracking.

4.4.4 Discussion of RQ4. In **RQ1**, we have shown that eye tracking has the precision to identify gaze behavior on individual gates through AOI analysis. However, participants' actions are hard to interpret from eye tracking alone. At the same time, TA has shortcomings because it is often difficult to interpret which gates participants are applying an action to, even when researchers use screen recordings as context for coding.

Our two use cases demonstrate that combining both methods can be valuable for interpreting problem-solving behavior in HRE. First, eye tracking allows us to spatially locate specific behaviors identified from TA. Second, eye tracking can be used to differentiate between eye-gaze patterns resulting from different high-level reverse engineering strategies. These techniques can reduce the amount of manual coding required in TA, thus providing a means to increase the sample size that can realistically be analyzed.

An important consideration here is the choice of CTA as the TA method, as eye tracking data cannot be easily synchronized with RTA protocols. In addition, the CTA method avoids the issue of participants not having a comprehensive memory of a task. However, the RTA method can be a valuable methodological adjunct when investigating challenging reverse engineering problems where participants are expected to become task saturated and therefore would be unable to talk while solving the task.

5 IMPLICATIONS AND OUTLOOK

In the following, we review the theoretical and practical relevance of our findings and outline future research directions. First, we highlight the broader applicability of our methodological approach beyond HRE. We then discuss our findings in relation to prior research on Think Aloud and eye tracking, and offer insight for improved combined analyses of both data sources. Second, our findings inform practitioners in enhancing hardware security or in developing educational resources for HRE. Finally, we suggest future research directions, including automated analyses to identify HRE problem-solving strategies and further investigating cognitive factors relevant to HRE.

5.1 Methodological Implications

Reverse engineering as an HCI phenomenon. HRE problem solving is not only visually demanding but also often involves navigating complex circuits. Lee and Johnson-Laird defined HRE as "the process of working out how to assemble components with known properties into a system that has the input–output relations of a target system," i. e., as a special kind of problem solving [43]. In particular, HRE problem solving requires frequent (re-)evaluation of assumptions as soon as a circuit's property – e. g., an input – changes. This phenomenon is not exclusive to hardware security, as

reverse engineering is a common problem-solving approach when people have to find and process information about a digital [12] or physical system [87] that is not intuitively accessible (e. g., not well designed), not given (e. g., expert knowledge), lost due to poor documentation, or even intentionally hidden (e. g., through obfuscation or dark patterns). Our methodological findings may therefore be used to inform studies in the above cases and may also be applied in various HCI contexts.

Methodological considerations in relation to prior work. We contextualize our methodological findings with previous work on methodological aspects of TA and eye tracking. Concerning **RQ2** (see Section 4.2), we discovered that RTA produces more overall codes but does not elicit significantly different information than CTA. Previous work has come to contradictory conclusions: Some papers find that RTA [57, 74] generates a higher number of codes, while other papers find that CTA [39, 78] produces a higher number of codes. Prior research tends to agree that RTA produces more insights into high-level reasoning and metacognitive reflections [39, 60, 74, 78][3], while participants in our study generated a substantial amount of high-level reflections during CTA as well. Second, our findings in **RQ3** (see Section 4.3) evidence that CTA does not affect task performance. Prior work either agrees with our results [21, 57] or reports either a small positive [60] or negative [42] effect on performance. Davies et al. [15] further report a notable skew in problem-solving behavior when participants are asked to verbalize. Considering a sample size of five participants per condition and an inconclusive result pertaining to the direction in which task behavior changes, this analysis should be interpreted cautiously. In **RQ3**, we further observed that CTA minimally prolongs eye fixations. In contrast, prior work by Prokop et al. [57] finds that fixation duration is significantly *shorter* when verbalizing. Ruckpaul et al. [60] generally agree, however, their observation does not reach statistical significance. Prokop et al.'s work also suggests that the differing findings could be a result of how participants concentrate on the tasks during verbalization [57], which is open to further investigation.

In summary, we find inconsistent results in the literature to date. We emphasize that the lack of concrete and widely applicable guidelines for combining TA and eye tracking necessitates a basic methodological evaluation – such as the one conducted in this paper – to ensure a sound experiment design.

Suggestions for Mixed-Methods Designs. In response to **RQ4**, we proposed two innovative semi-automated approaches to tightly combining eye tracking as a quantitative research method with qualitative content analysis of TA. Use Case I highlights how TA codes can be used to select and compare specific episodes of eye gazes. Use Case II suggests that eye tracking can help extract HRE problem-solving strategies more efficiently. While researchers have previously applied eye tracking and TA in the same experiment, some of this work uses eye tracking solely to provide a visual cue to participants during Retrospective Think Aloud [17, 78]. Other work often evaluates the results from both methods separately [25, 61]. Prior work that combines both sources either uses eye tracking for aligning AOIs or transcripts at a basic level [10, 27], filling silence periods in TA [18], or providing additional qualitative context for manual TA content analysis [14]. Our work expands on those approaches by integrating TA protocols with quantitative and automated analysis of eye gaze. We suggest that our joint analysis may offer more fine-grained insight into problem-solving behavior with reduced analysis effort, without extensive changes to existing experiment designs.

[3]Van Gog et al. conclude that *CTA* produces more insights on reasoning but base their conclusion on absolute numbers of codes alone. Taking the relative proportions into account, their findings appear to be generally in line with the other works.

5.2 Practical Applications

Enhancing hardware protection by cognitive obfuscation. A better understanding of HRE problem-solving processes may help to protect ICs against adversarial HRE, e. g., by competitors or hostile nation-state actors. While traditional obfuscation aims at defeating reverse engineering tools and algorithms [89], recent work introduced the concept of "cognitive obfuscation" [83]. This twist on the HCI framework attempts to hamper *human* understanding of a circuit. We observed that camouflaged and covert gates can draw considerable attention. Hardware designers may thus use those traditional obfuscated gates to introduce a false lead into an attacker's problem-solving process: By selectively obfuscating parts of the circuit unrelated to the security-critical areas, they may shift attackers' attention away from the relevant components. With this selective obfuscation, defenders can use obfuscated gates sparingly and thus economically while still wasting the attackers' time and resources. The eye tracking metrics introduced in our research could further be used to construct models that quantify the efficacy of such obfuscation.

Improving education in hardware security. Recent massive investments [20, 64] in domestic semiconductor fabrication are creating a major demand for the training of new talents in hardware security. In the field of HRE, where experts are already scarce, a serious shortage in hardware security educators thus arises. This demand for education may be supplemented with computer-aided tools for independent learning. Our method will motivate the design of learning content in a tutoring system for HRE novices. Specifically, a promising method is the application of Eye Movement Modeling Examples (EMMEs): Using eye tracking and TA, one captures experts' explanations as well as visualizations of their gaze locations as they solve an HRE problem. EMMEs foster learning by guiding attention, illustrating advanced perceptual strategies, and inducing a stronger social learning situation as learners watch experts performing a task [36, 37]. We consider EMMEs to be well-applicable to HRE education, following our evidence from Section 4.4.2 that differing netlist analysis strategies are indeed reflected in the corresponding eye gaze recordings.

5.3 Future Research Directions

Our work suggests multiple research directions towards a more comprehensive and less complex analysis of the human aspects of HRE, which can lead to more trusted hardware devices. We identify two major directions in the following.

Automating analysis of HRE strategies using eye tracking and TA. Section 4.4.2 demonstrates that our participants' problem-solving

strategies varied. We expect that manually identifying problem-solving strategies will become less feasible as the size and complexity of hardware circuits grow. Consequently, we suggest in future research to automate this process based on the following aspects. First, future approaches can further automate AOI definition, especially for dynamic visual stimuli (e. g., pop-up notices) in real-world HRE scenarios, which can be more challenging. Prior research in software engineering has shown that it is possible to define dynamic AOIs automatically from fixation saliency during software code navigation tasks [66]. Second, segmentation and labeling of eye tracking data can be streamlined with the aid of machine learning models tuned for coding and labeling the TA data, which will then enable more fine-grained analysis of the temporal phases in problem solving [16].

Studying the influences of cognitive factors within HRE problem solving. This work is exploratory with respect to understanding HRE processes based on participants' visual attention. HRE problem solving entails multiple sub-processes (see Section 4.2), which are influenced by different cognitive factors, such as prior knowledge and working memory [86]. These factors might be behind the varieties in problem solving that we have observed (see Section 4.4), which encourages a more granular analysis of eye-tracking data for these sub-processes and factors. Our case study in Section 4.4.1 exemplifies such potential, as fine-grained behaviors can be interpreted from just a few seconds of eye tracking data. By recruiting more participants and introducing additional study instruments, future studies could extend our methodology to quantifying the effect of different cognitive factors, such as working memory [4], in these sub-processes. Further, additional physiological data modalities could be analyzed jointly with eye movement, including pupil dilation, electroencephalogram, etc. Prior work indicates the benefits of such multi-modal data for understanding of human processes during screen interaction, e. g., with respect to spatial abilities [67, 72].

5.4 Conclusion

Understanding the human aspects of Hardware Reverse Engineering (HRE) is a crucial step in building more secure hardware. However, an in-depth understanding remains challenging through traditional methods, e. g., log file analysis, due to the complex problem-solving process of HRE. Recognizing visual processing as a key in people's HRE problem-solving process, we contribute an innovative mixed-methods HRE study that combines eye tracking and Think Aloud (TA) to gain deeper insight into such processes. Based on our study with 41 participants, we gathered evidence in support of the hypothesis that fixations are an appropriate eye-tracking metric to describe participants' visual attention within an HRE task. In particular, Area of Interest (AOI) analysis with fixation rates is valuable for quantifying attention to individual circuit elements. Furthermore, we evaluated two TA methods with eye tracking and identified both as suitable with distinct strengths in different applications. Based on two use cases, we demonstrated how eye tracking and TA complement each other for the analysis of HRE processes. From our results, we derive methodological implications that go beyond the specific domain of HRE and propose practical applications in the field of hardware security.

ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers for their constructive feedback on this work. We are also grateful to Sarah Naqvi, Jannik Schmöle, and Aurelia Sudjana, who were a great help in preparing the user study and in data analysis.

This work was supported by the PhD School "SecHuman – Security for Humans in Cyberspace" by the federal state of NRW, Germany, and the German Research Foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the States - EXC 2092 CASA - 390781972. This work was supported in part by the U.S. National Science Foundation under grants 1845469, 1942014, and 2003129.

We acknowledge the assistance of DALL·E 3 [53] in generating the artistic illustration of Figure 1.

CRediT authorship contribution statement. We describe each author's contributions with their initials below, following the Contributor Roles Taxonomy (CRediT) [2]. **Conceptualization**: J.L., S.B., C.W., Y.K., K.F., N.R., C.P. **Project administration:** R.W., J.L., S.B. **Investigation:** J.L., S.B., C.W. **Data Curation:** R.W., M.W., J.L. **Formal analysis:** R.W., M.W., J.L. **Writing – Original draft:** R.W., M.W., J.L., S.B. **Writing – Review & Editing:** R.W., M.W., J.L., S.B., M.E., Y.K., K.F., N.R., C.P. **Visualization:** R.W., M.W.

REFERENCES

[1] Nils Albartus, Max Hoffmann, Sebastian Temme, Leonid Azriel, and Christof Paar. 2020. DANA Universal Dataflow Analysis for Gate-Level Netlist Reverse Engineering. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2020, 4 (2020), 309–336. https://doi.org/10.13154/tches.v2020.i4.309-336

[2] Liz Allen, Alison O'Connell, and Veronique Kiermer. 2019. How can we ensure visibility and diversity in research contributions? How the Contributor Role Taxonomy (CRediT) is helping the shift from authorship to contributorship. *Learned Publishing* 32, 1 (2019), 71–74. https://doi.org/10.1002/leap.1210

[3] Leonid Azriel, Julian Speith, Nils Albartus, Ran Ginosar, Avi Mendelson, and Christof Paar. 2021. A Survey of Algorithmic Methods in IC Reverse Engineering. *Journal of Cryptographic Engineering* 11, 3 (2021), 299–315. https://doi.org/10.1007/s13389-021-00268-5

[4] Alan Baddeley. 1992. Working Memory. *Science* 255, 5044 (1992), 556–559. https://doi.org/10.1126/science.1736359

[5] Steffen Becker, Carina Wiesen, Nils Albartus, Nikol Rummel, and Christof Paar. 2020. An Exploratory Study of Hardware Reverse Engineering — Technical and Cognitive Processes. In *Sixteenth Symposium on Usable Privacy and Security, SOUPS 2020, August 7-11, 2020*. USENIX Association, Berkeley, CA, USA, 285–300. https://doi.org/10.1145/3577198

[6] Steffen Becker, Carina Wiesen, René Walendy, Nikol Rummel, and Christof Paar. 2023. ReverSim: A Game-Based Approach to Accessing Large Populations for Studying Human Aspects in Hardware Reverse Engineering. https://doi.org/10.48550/ARXIV.2309.05740 arXiv:2309.05740 [cs.CR]

[7] Roman Bednarik. 2012. Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations. *International Journal of Human-Computer Studies* 70, 2 (2012), 143–155. https://doi.org/10.1016/j.ijhcs.2011.09.003

[8] Shlomo Berkovsky, Ronnie Taib, Irena Koprinska, Eileen Wang, Yucheng Zeng, Jingjie Li, and Sabina Kleitman. 2019. Detecting personality traits using eye-tracking data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow, Scotland, GB, 1–12. https://doi.org/10.1145/3290605.3300451

[9] Gabriel Biehal and Dipankar Chakravarti. 1989. The Effects of Concurrent Verbalization on Choice Processing. *Journal of Marketing Research* 26, 1 (1989), 84. https://doi.org/10.2307/3172671

[10] Tanja Blascheck, Markus John, Steffen Koch, Leonard Bruder, and Thomas Ertl. 2016. Triangulating user behavior using eye movement, interaction, and think aloud data. In *Proceedings of the Ninth Biennial Symposium on Eye Tracking Research & Applications*. ACM, Charleston, SC, USA, 175–182. https://doi.org/10.1145/2857491.2857523

[11] S. Blythe, B. Fraboni, S. Lall, H. Ahmed, and U. de Riu. 1993. Layout Reconstruction of Complex Silicon Chips. *IEEE Journal of Solid-State Circuits* 28, 2 (1993), 138–145. https://doi.org/10.1109/4.192045

[12] Gerardo Canfora and Massimiliano Di Penta. 2007. New frontiers of reverse engineering. In *Future of Software Engineering (FOSE'07)*. IEEE Computer Society, Minneapolis, MN, USA, 326–341. https://doi.org/10.1109/FOSE.2007.15

[13] Ronald P. Cocchi, James P. Baukus, Lap Wai Chow, and Bryan J. Wang. 2014. Circuit Camouflage Integration for Hardware IP Protection. In *The 51st Annual Design Automation Conference 2014, DAC '14, San Francisco, CA, USA, June 1-5, 2014*. ACM, San Francisco, CA, USA, 153:1–153:5. https://doi.org/10.1145/2593069.2602554

[14] L. Cooke and E. Cuddihy. 2005. Using Eye Tracking to Address Limitations in Think-Aloud Protocol. In *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.* IEEE, Limerick, IE, 653–658. https://doi.org/10.1109/IPCC.2005.1494236

[15] Simon P. Davies. 1995. Effects of concurrent verbalization on design problem solving. *Design Studies* 16, 1 (1995), 102–116. https://doi.org/10.1016/0142-694x(95)90649-z

[16] Oliver Deane, Eszter Toth, and Sang-Hoon Yeo. 2023. Deep-SAGA: a deep-learning-based system for automatic gaze annotation from eye-tracking data. *Behavior Research Methods* 55, 3 (2023), 1372–1391. https://doi.org/10.3758/s13428-022-01833-4

[17] Fatma Elbabour, Obead Alhadreti, and Pam Mayhew. 2017. Eye Tracking in Retrospective Think-Aloud Usability Testing: Is There Added Value? *J. Usability Studies* 12, 3 (2017), 95–110. https://doi.org/10.5555/3190862.3190864

[18] Sanne Elling, Leo Lentz, and Menno De Jong. 2012. Combining Concurrent Think-Aloud Protocols and Eye-Tracking Observations: An Analysis of Verbalizations and Silences. *IEEE Transactions on Professional Communication* 55, 3 (2012), 206–220. https://doi.org/10.1109/TPC.2012.2206190

[19] K. A. Ericsson and H. A. Simon. 1993. *Protocol analysis: Verbal reports as data (Rev. ed.)*. The MIT Press, Cambridge, Massachusetts, USA. https://doi.org/10.7551/mitpress/5657.001.0001

[20] European Comission. 2022. A Chips Act for Europe – Comission Staff Working Document. https://digital-strategy.ec.europa.eu/en/library/european-chips-act-staff-working-document

[21] Jessica I. Fleck and Robert W. Weisberg. 2004. The use of verbal protocols as data: An analysis of insight in the candle problem. *Memory & Cognition* 32, 6 (2004), 990–1006. https://doi.org/10.3758/bf03196876

[22] Marc Fyrbiak, Sebastian Strauss, Christian Kison, Sebastian Wallat, Malte Elson, Nikol Rummel, and Christof Paar. 2017. Hardware Reverse Engineering: Overview and Open Challenges. In *IEEE 2nd International Verification and Security Workshop, IVSW 2017, Thessaloniki, Greece, July 3-5, 2017*. IEEE, Thessaloniki, GR, 88–94. https://doi.org/10.1109/IVSW.2017.8031550

[23] Marc Fyrbiak, Sebastian Wallat, Pawel Swierczynski, Max Hoffmann, Sebastian Hoppach, Matthias Wilhelm, Tobias Weidlich, Russell Tessier, and Christof Paar. 2019. HAL – The Missing Piece of the Puzzle for Hardware Reverse Engineering, Trojan Detection and Insertion. *IEEE Transactions on Dependable and Secure Computing* 16, 3 (2019), 498–510. https://doi.org/10.1109/TDSC.2018.2812183

[24] L.M. Galantucci, G. Percoco, G. Angelelli, C. Lopez, F. Introna, C. Liuzzi, and A. De Donno. 2006. Reverse engineering techniques applied to a human skull, for CAD 3D reconstruction and physical replication by rapid prototyping. *Journal of Medical Engineering & Technology* 30, 2 (2006), 102–111. https://doi.org/10.1080/03091900500131714

[25] Andreas Gegenfurtner and Marko Seppänen. 2013. Transfer of Expertise: An Eye Tracking and Think Aloud Study Using Dynamic Medical Visualizations. *Computers & Education* 63 (2013), 393–403. https://doi.org/10.1016/j.compedu.2012.12.021

[26] Joseph H. Goldberg and Jonathan I. Helfman. 2010. Comparing information graphics: a critical look at eye tracking. In *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*. ACM, Atlanta, GA, USA, 71–78. https://doi.org/10.1145/2110192.2110203

[27] Zhiwei Guan, Shirley Lee, Elisabeth Cuddihy, and Judith Ramey. 2006. The validity of the stimulated retrospective think-aloud method as measured by eye tracking. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, Montréal, Québec, CA, 1253–1262. https://doi.org/10.1145/1124772.1124961

[28] Ujjwal Guin, Ke Huang, Daniel DiMase, John M. Carulli, Mohammad Tehranipoor, and Yiorgos Makris. 2014. Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain. *Proc. IEEE* 102, 8 (2014), 1207–1228. https://doi.org/10.1109/JPROC.2014.2332291

[29] Roy S. Hessels, Diederick C. Niehorster, Chantal Kemner, and Ignace T. C. Hooge. 2017. Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC). *Behavior Research Methods* 49, 5 (2017), 1802–1823. https://doi.org/10.3758/s13428-016-0822-1

[30] Sture Holm. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics* 6 (1979), 65–70. http://www.jstor.org/stable/4615733

[31] Lida Huang, Mirjam Palosaari Eladhari, Sindri Magnússon, Hao Chen, and Ruijie Guo. 2023. Improving and Analyzing Sketchy High-Fidelity Free-Eye Drawing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. ACM, Pittsburgh, PA, USA, 856–870. https://doi.org/10.1145/3563657.3596121

[32] Stephen Hutt, Angela E.B. Stewart, Julie Gregg, Stephen Mattingly, and Sidney K. D'Mello. 2022. Feasibility of Longitudinal Eye-Gaze Tracking in the Workplace. *Proceedings of the ACM on Human-Computer Interaction* 6, ETRA (2022), 1–21. https://doi.org/10.1145/3530889

[33] Marcel A. Just and Patricia A. Carpenter. 1980. A theory of reading: from eye fixations to comprehension. *Psychological Review* 87, 4 (1980), 329–354. https://doi.org/10.1037/0033-295x.87.4.329

[34] Alexandros Kafkas and Daniela Montaldi. 2015. The pupillary response discriminates between subjective and objective familiarity and novelty. *Psychophysiology* 52, 10 (2015), 1305–1316. https://doi.org/10.1111/psyp.12471

[35] Philipp Kather, Rodrigo Duran, and Jan Vahrenhold. 2021. Through (Tracking) Their Eyes: Abstraction and Complexity in Program Comprehension. *ACM Transactions on Computing Education* 22, 2 (2021), 1–33. https://doi.org/10.1145/3480171

[36] Marie-Christin Krebs, Anne Schüler, and Katharina Scheiter. 2019. Just follow my eyes: The influence of model-observer similarity on Eye Movement Modeling Examples. *Learning and Instruction* 61 (2019), 126–137. https://doi.org/10.1016/j.learninstruc.2018.10.005

[37] Marie-Christin Krebs, Anne Schüler, and Katharina Scheiter. 2021. Do prior knowledge, model-observer similarity and social comparison influence the effectiveness of eye movement modeling examples for supporting multimedia learning? *Instructional Science* 49, 5 (2021), 607–635. https://doi.org/10.1007/s11251-021-09552-7

[38] Klaus Krippendorff. 2019. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, Inc., Thousand Oaks,CA, USA. https://doi.org/10.4135/9781071878781

[39] Hannu Kuusela and Pallab Paul. 2000. A Comparison of Concurrent and Retrospective Verbal Protocol Analysis. *The American Journal of Psychology* 113, 3 (2000), 387. https://doi.org/10.2307/1423365

[40] Chi Le, Vander Sloten Jos, Tan Hung Le, Khanh Lam, Shwe Soe, Nikolay Zlatov, Tam Phuoc Le, and Dinh Trung Pham. 2010. Medical reverse engineering applications and methods. , 186-196 pages. http://gala.gre.ac.uk/id/eprint/11735/

[41] Joy Yeonjoo Lee, Jeroen Donkers, Halszka Jarodzka, and Jeroen J.G. Van Merriënboer. 2019. How prior knowledge affects problem-solving performance in a medical simulation game: Using game-logs and eye-tracking. *Computers in Human Behavior* 99 (2019), 268–277. https://doi.org/10.1016/j.chb.2019.05.035

[42] N. Y. Louis Lee and P. N. Johnson-Laird. 2013. Strategic changes in problem solving. *Journal of Cognitive Psychology* 25, 2 (2013), 165–173. https://doi.org/10.1080/20445911.2012.719021

[43] N. Y. Louis Lee and P. N. Johnson-Laird. 2013. A Theory of Reverse Engineering and its Application to Boolean Systems. *Journal of Cognitive Psychology* 25, 4 (2013), 365–389. https://doi.org/10.1080/20445911.2013.782033

[44] Serena Lee-Cultura, Kshitij Sharma, Giulia Cosentino, Sofia Papavlasopoulou, and Michail Giannakos. 2021. Children's play and problem solving in motion-based educational games: Synergies between human annotations and multi-modal data. In *Interaction Design and Children*. Association for Computing Machinery, New York, NY, USA, 408–420. https://doi.org/10.1145/3459990.3460702

[45] Howard Levene et al. 1960. Robust tests for equality of variances. Contributions to probability and statistics. *Essays in honor of Harold Hotelling* 278 (1960), 292. https://doi.org/10.2307/2285659

[46] Hubert W. Lilliefors. 1967. On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown. *J. Amer. Statist. Assoc.* 62, 318 (1967), 399–402. https://doi.org/10.1080/01621459.1967.10482916

[47] Yu-Tzu Lin, Cheng-Chih Wu, Ting-Yun Hou, Yu-Chih Lin, Fang-Ying Yang, and Chia-Hu Chang. 2016. Tracking students' cognitive processes during program debugging—An eye-movement approach. *IEEE Transactions on Education* 59, 3 (2016), 175–186. https://doi.org/10.1109/TE.2015.2487341

[48] Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*. Springer, London, GB, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

[49] H. B. Mann and D. R. Whitney. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics* 18, 1 (1947), 50 – 60. https://doi.org/10.1214/aoms/1177730491

[50] Alessandro Mantovani, Simone Aonzo, Yanick Fratantonio, and Davide Balzarotti. 2022. RE-Mind: a First Look Inside the Mind of a Reverse Engineer. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Berkeley, CA, USA, 2727–2745. https://www.usenix.org/conference/usenixsecurity22/presentation/mantovani

[51] Travis Meade, Shaojie Zhang, and Yier Jin. 2016. Netlist Reverse Engineering for High-Level Functionality Reconstruction. In *21st Asia and South Pacific Design Automation Conference, ASP-DAC 2016, Macao, Macao, January 25-28, 2016*. IEEE, Macao, CN, 655–660. https://doi.org/10.1109/ASPDAC.2016.7428086

[52] Unaizah Obaidellah, Mohammed Al Haek, and Peter C.-H. Cheng. 2018. A survey on the usage of eye-tracking in computer programming. *Comput. Surveys* 51, 1 (2018), 1–58. https://doi.org/10.1145/3145904

[53] openai.com. 2024. DALL-E 3. https://openai.com/dall-e-3. [Online; accessed 2024-February-21].

[54] Oskar Palinko, Andrew L. Kun, Alexander Shyrokov, and Peter Heeman. 2010. Estimating cognitive load using remote eye tracking in a driving simulator. In *Proceedings of the 2010 symposium on eye-tracking research & applications*. ACM, New York, NY, USA, 141–144. https://doi.org/10.1145/1743666.1743701

[55] Sofia Papavlasopoulou, Kshitij Sharma, Michail Giannakos, and Letizia Jaccheri. 2017. Using eye-tracking to unveil differences between kids and teens in coding activities. In *proceedings of the 2017 conference on interaction design and children*. ACM, New York, NY, USA, 171–181. https://doi.org/10.1145/3078072.3079740

[56] Karl Pearson. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (July 1900), 157–175. https://doi.org/10.1080/14786440009463897

[57] Michal Prokop, Ladislav Pilar, and Ivana Tichá. 2020. Impact of Think-Aloud on Eye-Tracking: A Comparison of Concurrent and Retrospective Think-Aloud for Research on Decision-Making in the Game Environment. *Sensors* 20, 10 (2020), 2750. https://doi.org/10.3390/s20102750

[58] Endres Puschner, Thorben Moos, Steffen Becker, CChristian Kison, Amir Moradi, and Christof Paar. 2023. Red Team vs. Blue Team: A Real-World Hardware Trojan Detection Case Study Across Four Modern CMOS Technology Generations. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 56–74. https://doi.org/10.1109/SP46215.2023.00044

[59] M. G. Rekoff. 1985. On reverse engineering. *IEEE Transactions on Systems, Man, and Cybernetics* 15, 2 (1985), 244–252. https://doi.org/10.1109/TSMC.1985.6313354

[60] Anne Ruckpaul, Thomas Fürstenhöfer, and Sven Matthiesen. 2015. Combination of Eye Tracking and Think-Aloud Methods in Engineering Design Research. In *Design Computing and Cognition '14*. Springer International Publishing, Cham, DE, 81–97. https://doi.org/10.1007/978-3-319-14956-1_5

[61] L. Salmerón, J. Naumann, V. García, and I. Fajardo. 2017. Scanning and Deep Processing of Information in Hypertext: An Eye Tracking and Cued Retrospective Think-aloud Study. *Journal of Computer Assisted Learning* 33, 3 (2017), 222–233. https://doi.org/10.1111/jcal.12152

[62] Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, New York, NY, USA, 71–78. https://doi.org/10.1145/355017.355028

[63] Texplained SARL. 2021. Chipjuice IC Reverse Engineering Software. https://www.texplained.com/about-us/chipjuice-software/. [Online; accessed 2024-February-22].

[64] Senate of the United States. 2022. CHIPS and Science Act 2022 (P.L. 117-167). https://www.congress.gov/bill/117th-congress/house-bill/4346/text

[65] Bicky Shakya, Hao-Ting Shen, Mark Mohammad Tehranipoor, and Domenic Forte. 2019. Covert Gates: Protecting Integrated Circuits with Undetectable Camouflaging. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2019, 3 (2019), 86–118. https://doi.org/10.13154/tches.v2019.i3.86-118

[66] Zohreh Sharafi, Ian Bertram, Michael Flanagan, and Westley Weimer. 2022. Eyes on Code: A Study on Developers' Code Navigation Strategies. *IEEE Transactions on Software Engineering* 48, 5 (2022), 1692–1704. https://doi.org/10.1109/TSE.2020.3032064

[67] Zohreh Sharafi, Yu Huang, Kevin Leach, and Westley Weimer. 2021. Toward an objective measure of developers' cognitive activities. *ACM Transactions on Software Engineering and Methodology* 30, 3 (2021), 1–40. https://doi.org/10.1145/3434643

[68] Zohreh Sharafi, Timothy Shaffer, Bonita Sharif, and Yann-Gaël Guéhéneuc. 2015. Eye-tracking metrics in software engineering. In *2015 Asia-Pacific Software Engineering Conference (APSEC)*. IEEE Computer Society, New Dehli,IN, 96–103. https://doi.org/10.1109/APSEC.2015.53

[69] Zohreh Sharafi, Bonita Sharif, Yann-Gaël Guéhéneuc, Andrew Begel, Roman Bednarik, and Martha Crosby. 2020. A practical guide on conducting eye tracking studies in software engineering. *Empirical Software Engineering* 25 (2020), 3128–3174. https://doi.org/10.1007/S10664-020-09829-4

[70] Zohreh Sharafi, Zéphyrin Soh, and Yann-Gaël Guéhéneuc. 2015. A systematic literature review on the usage of eye-tracking in software engineering. *Information and Software Technology* 67 (2015), 79–107. https://doi.org/10.1016/j.infsof.2015.06.008

[71] Vasileios Skaramagkas, Giorgos Giannakakis, Emmanouil Ktistakis, Dimitris Manousos, Ioannis Karatzanis, Nikolaos S. Tachos, Evanthia Tripoliti, Kostas Marias, Dimitrios I. Fotiadis, and Manolis Tsiknakis. 2021. Review of eye tracking metrics involved in emotional and cognitive processes. *IEEE Reviews in Biomedical Engineering* 16 (2021), 260–277. https://doi.org/10.1109/RBME.2021.3066072

[72] Gino Slanzi, Jorge A. Balazs, and Juan D. Velásquez. 2017. Combining eye tracking, pupil dilation and EEG analysis for predicting web users click intention. *Information Fusion* 35 (2017), 51–57. https://doi.org/10.1016/j.inffus.2016.09.003

[73] Pramod Subramanyan, Nestan Tsiskaridze, Wenchao Li, Adrià Gascón, Wei Yang Tan, Ashish Tiwari, Natarajan Shankar, Sanjit A. Seshia, and Sharad Malik. 2014. Reverse Engineering Digital Circuits Using Structural and Functional Analyses. *IEEE Transactions on Emerging Topics in Computing* 2, 1 (2014), 63–80. https://doi.org/10.1109/TETC.2013.2294918

[74] K. Lynn Taylor and Jean-Paul Dionne. 2000. Accessing problem-solving strategy knowledge: The complementary use of concurrent verbal protocols and retrospective debriefing. *Journal of Educational Psychology* 92, 3 (2000), 413–425. https://doi.org/10.1037/0022-0663.92.3.413

[75] tobii.com. 2023. Tobii Pro Eye Tracker Manager. https://www.tobii.com/products/software/applications-and-developer-kits/tobii-pro-eye-tracker-manager#overview. [Online; accessed 2024-February-21].

[76] Randy Torrance and Dick James. 2009. The State-of-the-Art in IC Reverse Engineering. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Springer, Berlin, DE, 363–381. https://doi.org/10.1007/978-3-642-04138-9_26

[77] Hidetake Uwano, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto. 2006. Analyzing individual performance of source code review using reviewers' eye movement. In *Proceedings of the 2006 symposium on Eye tracking research & applications*. Association for Computing Machinery, New York, NY, USA, 133–140. https://doi.org/10.1145/1117309.1117357

[78] Tamara van Gog, Fred Paas, Jeroen J. G. van Merriënboer, and Puk Witte. 2005. Uncovering the Problem-Solving Process: Cued Retrospective Reporting Versus Concurrent and Retrospective Reporting. *Journal of Experimental Psychology: Applied* 11, 4 (2005), 237–244. https://doi.org/10.1037/1076-898x.11.4.237

[79] Mélodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2012. Wearable eye tracking for mental health monitoring. *Computer Communications* 35, 11 (2012), 1306–1311. https://doi.org/10.1016/j.comcom.2011.11.002

[80] Sebastian Wallat, Nils Albartus, Steffen Becker, Max Hoffmann, Maik Ender, Marc Fyrbiak, Adrian Drees, Sebastian Maaßen, and Christof Paar. 2019. Highway to HAL: Open-sourcing the First Extendable Gate-level Netlist Reverse Engineering Framework. In *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF 2019, Alghero, Italy, April 30 - May 2, 2019*, Francesca Palumbo, Michela Becchi, Martin Schulz, and Kento Sato (Eds.). ACM, New York, NY, USA, 392–397. https://doi.org/10.1145/3310273.3323419

[81] John B. Watson. 2009. Is thinking merely the action of language mechanisms? *British Journal of Psychology* 100, S1 (2009), 169–180. https://doi.org/10.1348/000712608x336095

[82] B. L. Welch. 1947. The Generalization of 'Student's' Problem When Several Different Population Variances are Involved. *Biometrika* 34, 1-2 (1947), 28–35. https://doi.org/10.1093/biomet/34.1-2.28

[83] Carina Wiesen, Nils Albartus, Max Hoffmann, Steffen Becker, Sebastian Wallat, Marc Fyrbiak, Nikol Rummel, and Christof Paar. 2019. Towards Cognitive Obfuscation: Impeding Hardware Reverse Engineering based on Psychological Insights. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference, ASPDAC 2019, Tokyo, Japan, January 21-24, 2019*, Toshiyuki Shibuya (Ed.). ACM, New York, NY, USA, 104–111. https://doi.org/10.1145/3287624.3288741

[84] Carina Wiesen, Steffen Becker, Nils Albartus, Christof Paar, and Nikol Rummel. 2019. Promoting the Acquisition of Hardware Reverse Engineering Skills. In *IEEE Frontiers in Education Conference, FIE 2019, Cincinnati, OH, USA, October 16-19, 2019*. IEEE, Cincinnati, OH, USA, 1–9. https://doi.org/10.1109/FIE43999.2019.9028668

[85] Carina Wiesen, Steffen Becker, Marc Fyrbiak, Nils Albartus, Malte Elson, Nikol Rummel, and Christof Paar. 2018. Teaching Hardware Reverse Engineering: Educational Guidelines and Practical Insights. In *IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE 2018, Wollongong, Australia, December 4-7, 2018*. IEEE, Wollongong, NSW, AU, 438–445. https://doi.org/10.1109/TALE.2018.8615270

[86] Carina Wiesen, Steffen Becker, René Walendy, Christof Paar, and Nikol Rummel. 2023. The Anatomy of Hardware Reverse Engineering: An Exploration of Human Factors During Problem Solving. *Comput.-Hum. Interact* 30, 4 (2023), 62:1–62:44. https://doi.org/10.1145/3577198

[87] Alexander Winnicki, Marina Krotofil, and Dieter Gollmann. 2017. Cyber-Physical System Discovery: Reverse Engineering Physical Processes. In *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security (ASIA CCS '17)*. ACM, New York, NY, USA, 3–14. https://doi.org/10.1145/3055186.3055195

[88] Daesub Yoon and N. Hari Narayanan. 2004. Mental imagery in problem solving: an eye tracking study. In *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2004, San Antonio, Texas, USA, March 22-24, 2004*. ACM, New York, NY, USA, 77–84. https://doi.org/10.1145/968363.968382

[89] Jiliang Zhang. 2015. A practical logic obfuscation technique for hardware security. *IEEE Transactions on very large scale integration (VLSI) systems* 24, 3 (2015), 1193–1197. https://doi.org/10.1109/TVLSI.2015.2437996

[90] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2019. Evaluation of appearance-based methods and implications for gaze-based applications. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300646

A FULL CODEBOOK

Main codes from Figure 11 are marked in **bold**.

- Strategies / Reasoning
  - **concluding actions / insight**
    - justification
    - elimination
      - partial solution
      - branch
      - irrelevant input
    - decamouflaging
    - unsuccsessful strategy
      - perseveres
    - reaching a conclusion
  - **theory crafting**
    - planning
    - theory / assumption / guess
      - speculation
    - validating
  - **backward tracking**
  - **forward tracking**
- **Exploration**
  - problem exceeds capabilities
  - starting point identification
  - localization of camouflaged gate
  - circuit exploration
  - goal state identification
- **Recall**
  - recalls prior insight
  - recognizes known sub-problem
- Errors and Error Correction
  - **corrective action**
    - identifying problem
    - realizing mistake
    - correcting mistake
  - **error**
    - incorrect reasoning
    - misinterpretation
    - forgetting
    - confusion
    - input error
- Misc
  - **self-talk**
    - self instruction
  - **external interruption**
  - **participant not talking**
  - **mumbles**
- RTA
  - **orientation**
  - **uncertainty / error**
  - **correction**
  - **insight**

B FIXATION DURATION

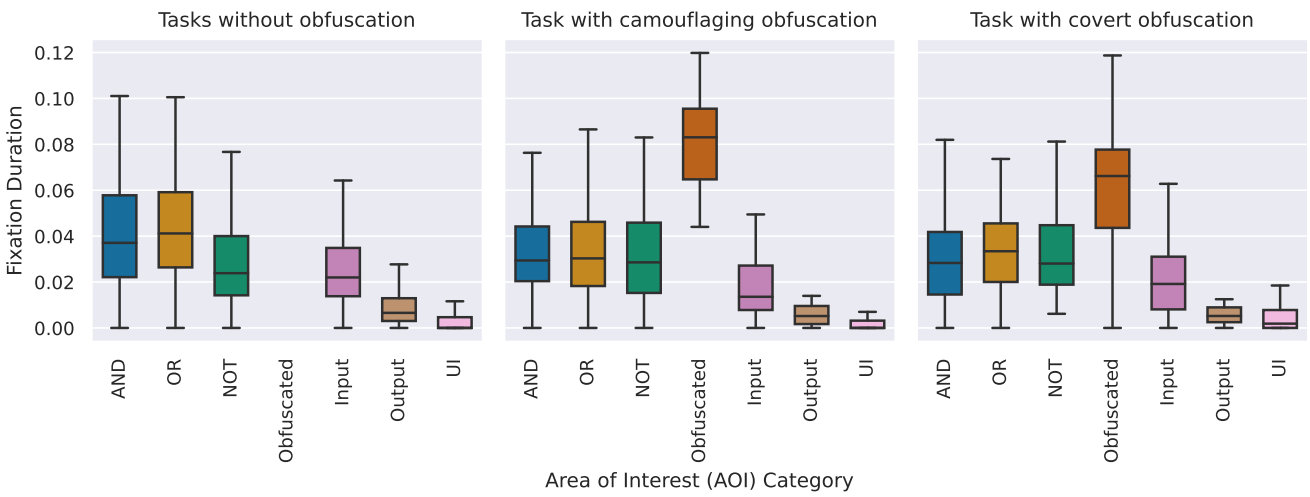


Figure 16: Statistics of fixation duration for each AOI category under different task complexities (RTA group). The basic logic gate types (AND, OR, NOT) receive similar attention across all types of the HRE tasks, but are outweighed by both camouflaged and covert gates. Output and UI elements receive little attention.