One Solves All: Exploring ChatGPT's Capabilities for Fully Automated Simple Power Analysis on Cryptosystems

Wenquan Zhou, An Wang, Yaoling Ding*, Congming Wei, Jingqi Zhang, Liehuang Zhu
School of Cyberspace Science and Technology, Beijing Institute of Technology
Beijing, China
Email: {zhouwenquan, wanganl, dyl19, weicm, zhangjq, liehuangz}@bit.edu.cn

***Abstract*—Side-channel analysis is a powerful technique to extract secret data from cryptographic devices. However, this task heavily relies on experts and specialized tools, particularly in the case of simple power analysis (SPA). Meanwhile, ChatGPT, a leading example of large language models, has attracted great attention and been widely applied for assisting users with complex tasks. Despite this, ChatGPT's capabilities for fully automated SPA, where prompts and traces are input only once, have yet to be systematically explored and improved. In this paper, we introduce a novel prompt template with three expert strategies and conduct a large-scale evaluation of ChatGPT's capabilities for SPA. We establish a dataset comprising seven sets of real power traces from various implementations of public-key cryptosystems, including RSA, ECC, and Kyber, as well as eighteen sets of simulated power traces that illustrate typical SPA leakage patterns. The results indicate that ChatGPT fails to be directly used for SPA. However, by applying the expert strategies, we successfully recovered the private keys for all twenty-five traces, which demonstrate that non-experts can use ChatGPT with our expert strategies to perform fully automated SPA.**

***Index Terms*—AI and Machine Learning, Security & Privacy, Test**

I. Introduction

Hardware and embedded systems security is essential, requiring cryptographic moudles to protect. Before entering the market, these products typically need experts and specialized tools to perform security tests. Recently, the swift advancement of AI technology has led to a remarkable surge in powerful large language models (LLMs). Leading global companies like OpenAI, Meta, and Google, along with numerous open-source contributors, have played a pivotal role in advancing the development of a wide range of LLMs. To date, LLMs have achieved significant success and found widely used in various domains, including code generation [1], vulnerability management [2], and anomaly detection [3]. Among the numerous LLMs, ChatGPT [4] has attracted widespread attention for its exceptional natural language processing and multimodal learning capabilities. Impressively, ChatGPT became the fastest-growing app worldwide, reaching 100 million users just two months after its launch.

In 1999, Kocher first proposed side-channel analysis for cryptosystems and successfully recovered the key using timing analysis and simple power analysis (SPA) [5]. Public-key cryptosystems such as RSA, ECC, and Kyber are widely used for identity recognition and digital signatures. These algorithms demonstrate distinctive power consumption patterns, rendering them vulnerable to SPA [6]. Consequently, SPA has become a crucial step for evaluating the security of cryptographic devices in standard [7]. Researchers have developed several effective SPA techniques by combining machine learning and mathematical analysis [8]. However, these methods still rely heavily on expert and specialized tools. Since ChatGPT can be applied in numerous fields, could it also be used as illustrated in Figure 1 to achieve fully automated SPA for public-key algorithms?

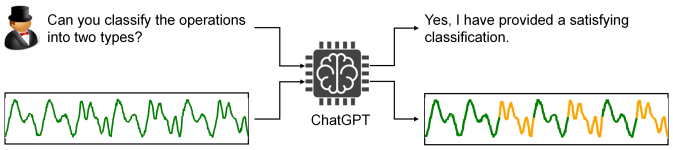


Figure 1. ChatGPT for SPA.

To fill this research gap, in this paper, we explore: **Can ChatGPT directly assist evaluators in performing fully automated SPA classification tasks for public-key cryptosystems?** Given the similarities between classification in SPA and anomaly detection tasks, applying ChatGPT to solve these tasks is highly plausible. Specifically, we aim to investigate whether GPT can directly perform fully automated public-key SPA. Besides, considering the impact of existing prompt engineering methods, we introduce a novel prompt template with three expert strategies to evaluate its effect on ChatGPT's capabilities. Finally, for the difficulties encountered by ChatGPT for SPA, we seek to shed light on aspects for future exploration.

To address the question, we established a dataset comprising seven sets of real power traces from various implementations of public-key cryptosystems, including RSA, ECC, and Kyber, as well as eighteen sets of simulated power traces representing common SPA leakage patterns. By leveraging this dataset, we evaluated ChatGPT's performance for SPA classification tasks by sending different prompts. We investigate the effect of each prompt by measuring the one-prompt success rate (OPSR),

*Yaoling Ding is the corresponding author.

which is defined as the proportion of correct classifications achieved over ten independent trials, with each trial inputting a prompt only once. Then, we investigate the impact of different prompt engineering strategy. Finally, we analyzed ChatGPT's responses to identify bottlenecks for each task.

Our evaluation and analysis results demonstrate that (1) ChatGPT cannot be directly used for SPA. (2) By applying expert strategies, we successfully recovered the private keys for all twenty-five traces, which demonstrate that non-experts can use ChatGPT with our expert strategies to perform fully automated SPA. (3) Intuitively, the more information provided in the prompt, the better ChatGPT performs. However, our investigation reveals that providing excessive information to ChatGPT can lead to memory loss and hallucinations. Therefore, directing ChatGPT to prioritize relevant and constructive information over potentially problematic content is a critical area for further research. Our contributions are as follows.

- We conduct the first large-scale evaluation of ChatGPT for SPA tasks. The results indicate that ChatGPT cannot be directly used for SPA.
- We introduce a novel prompt template with three expert strategies. By applying our strategies, we successfully recovered the private keys for all twenty-five traces, which demonstrate that non-experts can use ChatGPT with our expert strategies to perform fully automated SPA.
- We uncover the bottlenecks encountered by ChatGPT for SPA and shed light on promising future directions to improve ChatGPT's performance.

The remainder of the paper is organized as follows. Section II gives the background about SPA and ChatGPT. Section III describes the research pipeline and expert strategies design. Section IV shows the experimental setup and evaluation results. Section V discusses the bottlenecks and future directions. Section VI concludes the paper.

II. Background

A. SPA on Public-key Cryptosystems

Currently, there are two commonly used public-key cryptosystems: RSA [9] and ECC [10], [11]. The sequence of cryptographic operations is directly related to the private key. Taking RSA as an example, if a private key bit is "0", the algorithm only performs modular square. However, if a private key bit is "1", the algorithm sequentially performs modular square and modular multiplication. Due to the typical differences in the execution operations for different bit values, these variations are often significantly reflected in power traces. Therefore, the core of SPA typically consists in classifying waveforms to different cryptographic operations to obtain information about private keys.

B. ChatGPT and Prompt

ChatGPT is a large language model developed by OpenAI with powerful language understanding and generation capabilities. Users can use ChatGPT through its web interface or official API [4]. A common method to adapt general models to specific tasks is model fine-tuning. However, this approach is often eschewed due to its labor-intensive requirements and significant resource consumption. As a result, attention has shifted toward optimizing the prompt, i.e., the input of ChatGPT, which significantly influences the relevance and accuracy of ChatGPT's output [12].

Currently, various prompt construction strategies are employed to enhance ChatGPT's capabilities. Among these, in-context learning has become a dominant paradigm [13]. The foundational approach to in-context learning, known as 0-shot prompting, instructs ChatGPT by directly describing the task and question [14]. However, this approach may struggle with unfamiliar tasks. To address this problem, researchers devised advanced prompts by integrating demonstrations. Depending on the volume of demonstration examples within a prompt, they can be classified as 1-shot prompting (with a singular demonstration example) or few-shot prompting (incorporating multiple demonstration examples) [15]. While well-structured demonstrations have proven effective for simple tasks, they tend to be less effective for intricate tasks. To counter these challenges, another line of works enhance ChatGPT by refining demonstration formats. This includes providing supplemental general information, such as role definitions [14], [16] and step-by-step thinking guidance [12].

III. Methodology

A. Research Pipeline

Figure 2 shows the pipeline of our research, which includes three phases: ① template design and dataset preparation, ② expert strategies design and optimization, and ③ large-scale evaluation.

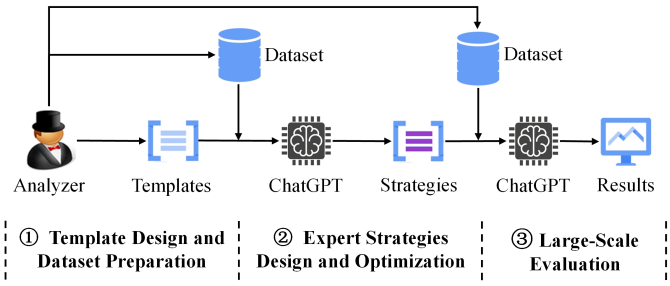


Figure 2. Research pipeline.

Currently, automatic prompt generation [17] is an ongoing research work that has not been well addressed. Consequently, in phase ①, according to the construction rules outlined in II-B, we first design three prompt templates listed in Table 1 manually based on the heuristics derived from existing widely adopted strategies [15]. Subsequently, We have established a power trace dataset which list in Table 2 and Table 3.

In phase ②, we first design many different expert strategies. Subsequently, we refine the strategies based on our manual analysis of ChatGPT's responses to limited traces in the dataset. Details regarding the strategies design and optimization process are discussed in III-B. As a result, we

Table 1. Three prompt templates.

Template Name	Template	Description
0-shot	<input><task description>	**Input**: The file contains a power trace from an RSA signing process, where each segment represents an operation. **Task Description**: Please identify these operations, which can be classified into two types: S and M.
general-info	<role><reinforce><input> <task description><zero-CoT>	**Role**: I would like you to act as an expert in side-channel analysis and signal processing, helping me analyze vulnerabilities in an encryption system. **Reinforce**: During the analysis process, please do not ask me any questions; instead, proceed step by step until you provide the final answer. **Input** & **Task Description**: (...Same as Above...) **Zero-CoT**: To complete this task, you may need to follow these steps: 1. Choose an appropriate method to segment the trace. 2. Choose an appropriate method to classify the operations into two types, S and M.
expertise	<role><reinforce><input> <task description><zero-CoT> <expert strategies>	**Role** & **Reinforce** & **Input** & **Task Description** & **Zero-CoT**: (...Same as Above...) **Expert Strategies**: (A selection of three expert strategies.)

acquire three expert strategies: **Preprocessing**, **Classification** and **Rectification**.

Finally, in phase ③, to fully explore ChatGPT's capabilities, we use 0-shot, general-info, and expertise prompts with a selection of expert strategies developed in phase ② to conduct a large-scale evaluation on the dataset.

B. Expert Strategies Design and Optimization

When humans get the power traces to do SPA, the approach is generally: (1) First segment the trace according to the known bit length of the private key [8], [18]. (2) Then apply moving average, filtering or other preprocessing operations to the segmented waveforms [19]. (3) Then choose a classification method such as evaluating visual information, or reducing the dimension followed by clustering to divide the waveforms into two categories [20], [21]. (4) Then check the results by certain rules. For example, the number of modular squares must be consistent with the known bit length of the private key under RSA algorithm without special SPA protection [22]. There must never be two consecutive modular multiplication operations. If results violate a rule, try a different classification method until the rule is no longer violated. From the perspective of human thinking logic, we refine (2) - (4) into three expert strategies when the private key of a public-key cryptosystems is n bits (where n represents the bit length of the private key) and disregarding unexpected segmentation issues. To ensure the effectiveness of each expert strategies, we assess them using limited traces in the dataset. Subsequently, we optimize the strategies based on our manual analysis of ChatGPT's responses. Finally, the three expert strategies are as follows:

- **Preprocessing**: Apply moving average, filtering or other preprocessing operations to the segmented waveforms.
- **Classification**: Please evaluate the classification by considering both numerical data and visual information from the waveform. For the visual aspect, you can use techniques such as shape recognition, peak count, or image-based anomaly detection methods. For the numerical data, consider dimensionality reduction followed by clustering to divide the operations into two categories.
- **Rectification**: There must be exactly n S operations, the first operation must be a S operation, and there must be at least one S operation between any two M operations. After completing the classification, please check the number of S operations and whether the M operations meet the spacing requirements. If the criteria are not met, please try a different classification method.

As shown in Figure 3, the expertise prompt differs from the general-info prompt by providing **Preprocessing** and **Classification** expert strategies. After removing the pink text, the rest represents the general-info prompt.

```
Role            I would like you to act as an expert in side-channel
                analysis and signal processing, helping me analyze
                vulnerabilities in an encryption system. During the analysis
Reinforce       process, please do not ask me any questions; instead,
                proceed step by step until you provide the final answer.
Input           The file contains a power trace from an RSA signing
                process, where each segment represents an operation.
Task            Please identify these operations, which can be classified
Description     into two types: S and M.
                To complete this task, you may need to follow these
                steps:
Zero-CoT          1. Choose an appropriate method to segment the trace
                into 36 parts.
                  2. First apply moving average, filtering or other
                preprocessing operations to the segmented waveforms,
                then choose an appropriate method to classify the
                operations into two types, S and M.
                  Please evaluate the classification by considering both
                numerical data and visual information from the waveform.
Expert          For the visual aspect, you can use techniques such as
Strategies      shape recognition, peak count, or image-based anomaly
                detection methods. For the numerical data, consider
                dimensionality reduction followed by clustering to divide
                the operations into two categories.
```

Figure 3. An example of the expertise prompt for side-channel analysis.

IV. Evaluation Results

In this section, we elaborate on the evaluation results of ChatGPT for SPA tasks. We seek the answers to the research

question proposed in Section I by using OPSR, investigating the impact of different expert strategies, and exploring the potential future research directions to address the bottlenecks encountered by ChatGPT.

A. Setup

We use different accounts to access ChatGPT-4 through OpenAI's web interface. Each experiment started a new conversation and turn off the memory function to ensure that each experiment was repeated independently. We have established a dataset[1] with seven sets of real power traces from different implementations of public-key cryptosystems, including RSA, ECC, and Kyber, as well as eighteen sets of simulated power traces illustrating common SPA leakage patterns. Table 2 and Table 3 presents the key information about the traces in the dataset.

The real dataset includes four different implementations of RSA, two implementations of ECC, and one implementation of the post-quantum algorithm Kyber [23]. To differentiate between devices, we assigned symbolic abbreviations, which are subsequently used throughout the paper. Details such as private-key lengths ($\boldsymbol{L}_{\text{key}}$), number of points, and other key information are described in Table 2. And ECC-AT is a toy implementation due to limited memory and computational resources on the AT89S52. During the experiment, in light of ChatGPT's response limitations, we provided extracted traces containing thirty-six operations per submission to ChatGPT.

Table 2. The dataset of public-key cryptosystems.

Algorithm	$\boldsymbol{L}_{\text{key}}$	Operations	Device	Implementation
RSA	1024	1562	smart card (SC)	co-design
RSA	1024	1536	ASIC (AS)	hardware
RSA	1024	1531	SAKURA-G (SG)	hardware
RSA	1024	1535	STM32F429 (S9)	software
ECC	128	192	AT89S52 (AT)	software
ECC	256	372	smart card (SC)	co-design
Kyber	256	256	STM32F407 (S7)	software

The simulated dataset includes six distinct categories, each subdivided into three levels, differentiated by Gaussian noise with standard deviations of 0.01, 0.1, and 0.2, denoted as σ_1, σ_{10}, and σ_{20}, respectively. Furthermore, we apply the modular square (S) and modular multiplication (M) operations of the RSA algorithm as examples, characterizing the signal-to-noise ratio (SNR) difference between two operations traces by calculating the Euclidean distance between them:

$$d_{\text{Euclidean}} = \sqrt{\sum_{t=1}^{N} \left(y_1(t) - y_2(t)\right)^2} \tag{1}$$

where $d_{\text{Euclidean}}$ denotes the Euclidean distance between two operations, t denotes the t-th point in the operation, and N represents the total number of points in each operation. $y_1(t)$ and $y_2(t)$ are the values of the trace at point t.

[1]https://github.com/haillife/One-Solves-All

We use the same $d_{\text{Euclidean}}$ between two operations in the first five simulated traces:

- Simulated-1: S and M is different overall.
- Simulated-2: M is slightly higher than S overall.
- Simulated-3: M differs from S in only one point.
- Simulated-4: M has 10 discrete points differ from S.
- Simulated-5: M has 10 consecutive points differ from S.

In Simulated-1 to Simulated-5, S and M share the same number of points 100. In Simulated-6, M contains ten additional points compared to S, but the overall shape remains identical.

Table 3. The dataset of simulated traces.

Name	$\boldsymbol{L}_{\text{key}}$	Points	Noise
Simulated-1	24	3600	$\sigma_1, \sigma_{10}, \sigma_{20}$
Simulated-2	24	3600	$\sigma_1, \sigma_{10}, \sigma_{20}$
Simulated-3	24	3600	$\sigma_1, \sigma_{10}, \sigma_{20}$
Simulated-4	24	3600	$\sigma_1, \sigma_{10}, \sigma_{20}$
Simulated-5	24	3600	$\sigma_1, \sigma_{10}, \sigma_{20}$
Simulated-6	24	3720	$\sigma_1, \sigma_{10}, \sigma_{20}$

By leveraging dataset, we evaluated ChatGPT's capabilities for SPA tasks by sending 0-shot, general-info, and expertise prompts with different expert strategies. We label the three expert strategies Preprocessing, Classification, and Rectification, as 1, 2, and 3, respectively. For example, expertise-2&3 represents an expertise prompt that uses the Classification and Rectification strategies. And we investigate the effect of each prompt by measuring the one-prompt success rate (OPSR), which is defined as the proportion of correct classifications achieved over ten independent trials, with each trial inputting a prompt only once.

Table 4. Evaluation results of real dataset.

Prompt	RSA				ECC		Kyber
	SC	AS	SG	S9	AT	SC	S7
0-shot	0	0	0	0.1	0	0	0
general-info	0	0	0.17	0.4	0	0	0
expertise-1	0.1	0	0	0.14	0	0	0
expertise-2	0.1	0	0	0.125	0.125	0.25	0.1
expertise-3	0.2	0	0	0	0	0.3	–
expertise-1&2	0	0	0	0.5	**0.25**	0	**0.5**
expertise-1&3	0	0	0	0.4	0	0	–
expertise-2&3	**0.25**	0.125	0	**0.9**	0	**0.9**	–
expertise-1&2&3	0.2	**0.5**	**0.2**	0.4	0.14	0.14	–

B. Results

Table 4 and Table 5 reports the results of evaluation and – represents the expert strategy is not suitable.

*a) **ChatGPT cannot be directly used for SPA**:* For comparison, we first evaluate ChatGPT's capabilities by 0-shot, general-info, and expertise with strategies. The OPSR results from the analysis of seven real and eighteen simulated traces, as detailed in Table 4 and Table 5, indicate that ChatGPT cannot be directly used for SPA, achieving nearly

Table 5. Evaluation results of simulated dataset.

Prompt	Simulated-1			Simulated-2			Simulated-3			Simulated-4			Simulated-5			Simulated-6		
	σ_1	σ_{10}	σ_{20}	σ_1	σ_{10}	σ_{20}	σ_1	σ_{10}	σ_{20}	σ_1	σ_{10}	σ_{20}	σ_1	σ_{10}	σ_{20}	σ_1	σ_{10}	σ_{20}
0-shot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
general-info	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0.2	0	0	0	0	0
expertise-1	0	0	0	0.2	0.9	0.5	0.9	0	0	0	0	0	0	0	0	0	0	0
expertise-2	0.5	0.5	0.5	0.5	0.17	0.5	**1**	0	0	**1**	0.33	0	0.5	0.14	0	0	0	0
expertise-3	0.9	0	0	0.33	0	0	0	0	0	0.33	0	0	0	0	0	0	0	0
expertise-1&2	0.2	0.5	0.5	**1**	**1**	0.5	0.3	**1**	0	0.5	0	0	**1**	0	0	**0.5**	0.14	0
expertise-1&3	0.3	0	0.2	0	0	0	0.3	0	0	0.17	0	0	0.3	0	0	0	0	0
expertise-2&3	0.5	0	0.8	**1**	0.5	**1**	0.2	0	0	**1**	0.13	**0.5**	**1**	0.13	**0.25**	0.3	0.5	0
expertise-1&2&3	**1**	**1**	**1**	**1**	**1**	0.25	**1**	0.5	**0.5**	0.5	**0.5**	0	0.5	**0.25**	0	**0.5**	**0.8**	**0.4**

zero success in both 0-shot and general-info input. This lack of effectiveness is evident, as most traces yielded no successful analyses across ten attempts, with only a few displaying minimal success rates. However, capabilities significantly improve when prompts with expert strategies are employed, enabling successful private-key recovery of all twenty-five traces. Notably, the success rates fluctuate depending on the strategies applied, indicating that the effectiveness of these strategies can vary independently.

b) ***ChatGPT's capabilities****:* The OPSR results of the simulated traces with Gaussian noise of σ_{10} are detailed in table 5. It reveals that ChatGPT demonstrates a notable proficiency in distinguishing between Simulated-1 and Simulated-2 traces which share the same SNR leakage. While Simulated-1 can be distinguished by the naked eye, Simulated-2 remains visually indistinguishable. This finding suggests that ChatGPT holds an advantage over human visual analysis. Further analysis extends this observation to a broader range of leak scenarios within Simulated-3, Simulated-4, and Simulated-5. In these cases, ChatGPT effectively manages SPA tasks across varying leak types, including single-point leaks (Simulated-3), discrete multi-point leaks (Simulated-4), and continuous multi-point leaks (Simulated-5). Additionally, the results for Simulated-6 underscore ChatGPT's effectiveness in handling SPA tasks involving temporal leaks, reaffirming its utility in complex scenarios where precise timing information is essential. Collectively, these outcomes highlight ChatGPT's robustness and versatility in SPA tasks across various contexts and leakage categories.

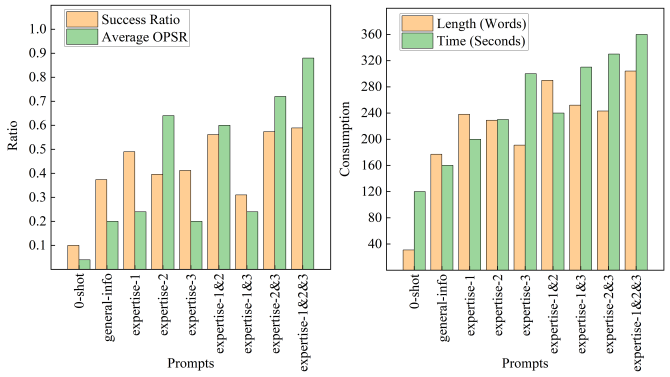
We examines the impact of Gaussian noise on the OPSR across various simulated datasets, confirms the anticipated influence of noise on signal processing tasks. The results demonstrate a clear trend: OPSR decreases as noise levels rise, aligning with expectations that higher noise generally impairs signal analysis success rates due to increased distortion and reduced clarity. Despite the anticipated decline in performance with increased noise levels, ChatGPT demonstrates substantial robustness and retains a notable degree of resistance to interference. This suggests that ChatGPT can effectively manage SPA tasks even under suboptimal conditions where noise impacts the data analysis process. Its ability to perform successfully across varying noise levels highlights both its adaptability and potential utility in environments where noise is an unavoidable factor.

c) ***Suggestions****:* The success ratio in Figure 4a indicates the number of successfully recovered traces out of the total twenty-five, while the average OPSR represents the mean OPSR of these successful recoveries. Figure 4a provides a comprehensive analysis of OPSR across different expertise prompts for SPA tasks, offering a deeper understanding of the relative effectiveness of various strategies when applied individually or in combination. The analysis indicates that expertise-1, when used alone, generally results in low OPSR, suggesting limited efficacy as a standalone approach. Specifically, in scenarios such as RSA-SC and Simulated-2, where leakage is represented by a single point, expertise-1 may actually hinder capability. This effect likely arises from ChatGPT's tendency under expertise-1 to inadvertently filter out essential signal points. Further data shows that combining strategies often improves OPSR, though more strategies (e.g., expertise-1&2&3) do not necessarily produce better outcomes than selective applications (e.g., expertise-1&2). This outcome may stem from ChatGPT's constraints in context memory, where an excess of strategies can lead to issues like memory loss and hallucinations. Additionally, expertise-3 demonstrates limitations in specific contexts, such as the kyber traces, where signals distinguish between consecutive bits (e.g., two consecutive "1" bits). In these cases, expertise-3 may be less effective or even unsuitable, indicating a need for adjustments or the potential exclusion of this strategy.

In Figure 4b, "Length" represents the total word count in this strategy, while "Time" denotes ChatGPT's average response time. This figure reveals that as the number of strategies increases, both the token count and response time also rise, though both remain within acceptable limits. In summary, the statistical findings suggest that expertise-2&3 is generally sufficient and effective in most cases. While using expertise-1&2&3 may yield a marginally higher average OPSR, the nuanced performance of each strategy combination under varying conditions advocates for a more tailored approach to optimize ChatGPT's capabilities.

V. Discussion

In this section, we uncover the bottlenecks encountered by ChatGPT for SPA and shed light on promising future



(a) Success ratio of prompts. **(b)** Consumption of prompts.

Figure 4. Success ratio and consumption of prompts.

directions to improve ChatGPT's performance.

Segmentation problem. The quality of trace segmentation directly impacts the accuracy of segment classification, which in turn affects private key recovery [22]. Currently, the most common method of segmentation are equidistant segmentation [8] and peak-based segmentation [18]. In this paper, we find that ChatGPT generally uses a straightforward equal-division strategy for segmentation, which is often effective. When the segments are not of equal length, we can guide ChatGPT to use the *find_peaks* function from the SciPy library, which identifies peaks in traces, to achieve successful segmentation. Therefore, we do not focus on segmentation in particular. We acknowledge that segmentation can become complex in certain cases, in such instances, our approach is to separate segmentation from classification tasks. In future work, we will further explore and enhance ChatGPT's capabilities in trace segmentation and preprocessing.

Prompting techniques. We manually constructed prompt templates based on prior works in LLM evaluation [24] and our empirical analysis. This manual approach is adopted due to the inherent challenges of automatic prompt engineering, a complex area that holds potential for stimulating research [17]. We also tried to design other expert strategies, such as providing examples. However, due to the significant length of traces and absence of a golden sample for SPA, this approach proved challenging. ChatGPT struggles to effectively learn features when the context exceeds its length limit, leading us to ultimately abandon this strategy. Nevertheless, developing additional expert strategies remains an interesting direction for future research.

Alternative AI approaches. We primarily evaluates ChatGPT's performance, given its prominence as the leading AI tool currently available. However, our prompt templates and evaluation pipeline are broadly applicable to other LLMs. Future work will entail comprehensive assessments of additional LLMs to rigorously investigate and benchmark their performance. Furthermore, exploring alternative AI methodologies, such as fine-tuning open-source models, may help mitigate certain identified limitations, thus offering a valuable avenue for future research.

Hallucination issues. We identify several instances of hallucination and implement mitigation strategies, including expert-driven prompts and meticulous manual verification. Nonetheless, effectively addressing hallucination in LLMs remains an unresolved challenge that warrants further investigation in subsequent research.

VI. Conclusion

In this paper, we conduct the first large-scale evaluation to explore ChatGPT's capabilities for SPA. Specifically, we propose a novel prompt template with three expert strategies. By sending the prompt to ChatGPT only once, we investigate ChatGPT's SPA capabilities using seven sets of real power traces from different implementations of public-key cryptosystems, including RSA, ECC, and Kyber, as well as eighteen sets of simulated power traces illustrating common SPA leakage patterns. The results indicate that ChatGPT cannot be directly used for SPA. However, by applying our strategies, we successfully recovered the private keys for all twenty-five traces, which demonstrate that non-experts can use ChatGPT with our expert strategies to perform fully automated SPA. Furthermore, we identify specific challenges encountered by ChatGPT and shed light on future research aimed at optimizing ChatGPT's capabilities for SPA.

References

[1] M. C. Tol and B. Sunar, "Zeroleak: Automated side-channel patching in source code using llms," in *European Symposium on Research in Computer Security*. Springer, 2024, pp. 290–310.

[2] P. Liu *et al.*, "Exploring ChatGPT's capabilities on vulnerability management," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 811–828.

[3] S. Alnegheimish, L. Nguyen, L. Berti-Equille, and K. Veeramachaneni, "Can large language models be anomaly detectors for time series?" in *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2024, pp. 1–10.

[4] ChatGPT, "ChatGPT," https://chatgpt.com, accessed: October 2024.

[5] P. Kocher, "Differential power analysis," in *Proc. Advances in Cryptology (CRYPTO'99)*, 1999.

[6] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks of modular exponentiation in smartcards," in *Cryptographic Hardware and Embedded Systems: First InternationalWorkshop, CHES'99 Worcester, MA, USA, August 12–13, 1999 Proceedings 1*. Springer, 1999, pp. 144–157.

[7] *Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules*, International Organization for Standardization Std. ISO/IEC 17 825, 2024.

[8] J. Heyszl, S. Mangard, B. Heinz, F. Stumpf, and G. Sigl, "Localized electromagnetic analysis of cryptographic implementations," in *Topics in Cryptology–CT-RSA 2012: The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27–March 2, 2012. Proceedings*. Springer, 2012, pp. 231–244.

[9] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[10] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[11] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the theory and application of cryptographic techniques*. Springer, 1985, pp. 417–426.

[12] B. Clavié, A. Ciceu, F. Naylor, G. Soulié, and T. Brightwell, "Large language models in the workplace: A case study on prompt engineering for job type classification," in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2023, pp. 3–17.

[13] Q. Dong *et al.*, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[14] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining zero-shot vulnerability repair with large language models," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 2339–2356.

[15] S. Gao, X. Wen, C. Gao, W. Wang, H. Zhang, and M. R. Lyu, "What makes good in-context demonstrations for code intelligence tasks with llms?" in *38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023, Luxembourg, September 11-15, 2023*. IEEE, 2023, pp. 761–773.

[16] C. S. Xia, Y. Wei, and L. Zhang, "Automated program repair in the era of large pre-trained language models," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, pp. 1482–1494.

[17] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 4222–4235.

[18] A. Wang, S. He, C. Wei, S. Sun, Y. Ding, and J. Wang, "Using convolutional neural network to redress outliers in clustering based side-channel analysis on cryptosystem," in *Smart Computing and Communication - 7th International Conference, SmartCom 2022, New York City, NY, USA, November 18-20, 2022, Proceedings*, ser. Lecture Notes in Computer Science, vol. 13828. Springer, 2022, pp. 360–370.

[19] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker, "Improving differential power analysis by elastic alignment," in *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6558. Springer, 2011, pp. 104–119.

[20] R. Specht, J. Heyszl, M. Kleinsteuber, and G. Sigl, "Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements," in *Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 9064. Springer, 2015, pp. 3–19.

[21] J. Heyszl, A. Ibing, S. Mangard, F. D. Santis, and G. Sigl, "Clustering algorithms for non-profiled single-execution attacks on exponentiations," in *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 8419. Springer, 2013, pp. 79–93.

[22] Z. Wang, Y. Ding, A. Wang *et al.*, "SPA-GPT: general pulse tailor for simple power analysis based on reinforcement learning," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, no. 4, pp. 40–83, 2024.

[23] Z. Xu, O. Pemberton, S. S. Roy, D. Oswald, W. Yao, and Z. Zheng, "Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber," *IEEE Transactions on Computers*, vol. 71, no. 9, pp. 2163–2176, 2021.

[24] W. Ma *et al.*, "The scope of chatgpt in software engineering: A thorough investigation," *CoRR*, vol. abs/2305.12138, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.12138