Poster Session Proceedings

9th IEEE European Symposium on Security and Privacy

IEEE
EURO
S&P
2024

Preface

This volume contains the abstracts of the posters accepted and presented at the EuroS&P 2024 conference, which was held on July 8-12, 2024, in Vienna. Following the tradition initiated with the 2022 edition of the conference, these proceedings are published on Zenodo (https://zenodo.org), an open research repository operated by CERN, which enables sharing and preserving of research outputs.

The poster session of Euro S&P is conceived as an opportunity for security and privacy researchers to share their recent results, and to obtain valuable feedback on their ongoing work from participants at the conference. In total, 10 abstracts were submitted to the poster session, which underwent a lightweight review process. In particular, reviews were aimed at checking the coherence of the abstracts with the scope of Euro S&P, rather than providing in-depth technical feedback as for a regular submission. Each abstract was reviewed by at least two members of the program committee. After the review phase, the program committee decided to accept all submitted abstracts.

We would like to thank all the authors for the creativity and effort that went into their submissions, and especially the poster presenters (including the invited posters' presenters), for their tireless engagement with the attendees during the lively poster session. We are also grateful to Edgar Weippl and Matteo Maffei, the general chairs; Sebastian Schrittwieser and Victoria Buchsbaum, the publication chairs; Herbert Bos and Ben Stock, the program chairs; Sandra Wotawa and Yvonne Poul, the financial chairs; and to everyone else who helped facilitate the poster session. Finally, we are also grateful to Ivan Liang for setting up the HotCRP platform for managing the submission process, and to the web chair Sandra Wotawa for updating the website with the updates related to the call for posters.

Luca Mariot, University of Twente, the Netherlands,

Dipti K. Sarmah, University of Twente, the Netherlands,

IEEE European Symposium on Security and Privacy 2024 Poster Chairs

Program Committee

Poster Chairs

Luca Mariot, University of Twente, the Netherlands

Dipti K. Sarmah, University of Twente, the Netherlands

PC Members

Michele Carminati	Politecnico di Milano, Italy
Alessandro Erba	CISPA Helmholtz Center for Information Security, Germany
Claudio Ferretti	University of Milano-Bicocca, Italy
Christina Kolb	University of Twente, the Netherlands
Azqa Nadeem	University of Twente, the Netherlands
Martina Saletta	University of Bergamo, Italy
Fatih Turkmen	University of Groningen, the Netherlands

List of Accepted Posters

1. Evaluation of Membership Inference Attacks on LoRA with Differential Privacy
Takuya Higashi and Tsunato Nakai

2. Poster: Towards Extensible Memory Isolation for Low-End Microcontrollers
Marton Bognar and Jo Van Bulck

3. Poster: Privacy-Preserving Billing for Local Energy Markets
Eman Alqahtani and Mustafa A. Mustafa

4. Poster: Towards a Digital Payment System for the Constrained Internet of Things
Mikolai Gütschow and Matthias Wählisch

5. FedCM for Research and Education
Erwin Kupris, Tobias Hilbig, Thomas Schreck

6. Empirical Cybersecurity Investment Decision-Making: Bridging the Gap Between Intuition and Metric-Driven Strategy
Nadia Lorraine Niyonsaba, Abhishta Abhishta, Jeroen van der Ham-de Vos, Laura Spierdijk

7. Byzantine Fault Tolerant State Machine Replication for Non-deterministic Applications
Pedro Camponˆes, Diogo Tavares, Jo˜ao Magalh˜aes, Henrique Domingos

8. Security and Privacy Heterogeneous Environment for Reproducible Experimentation (SPHERE)
Jelena Mirkovic, David Balenson, Brian Kocoloski, David Choffnes, Daniel Dubois, Geoff Lawler, Chris Tran, Joseph Barnes, Yuri Pradkin, Terry Benzel, Srivatsan Ravi, Ganesh Sankaran, Alba Regalado, and Luis Garcia

9. Improved Federated Learning with Non-IID Data Using Foundation Models
Fatima Abacha, Sin G. Teo, Lucas C. Cordeiro, Mustafa A. Mustafa

10. Integrating a Secure Processing Environment in an IoT Operating System
Lena Boeckmann, Thomas C. Schmidt, Matthias W¨ahlisch

Poster: Evaluation of Membership Inference Attacks on LoRA with Differential Privacy

1st Takuya Higashi
Mitsubishi Electric Corporation
Kanagawa, Japan
Higashi.Takuya@da.MitsubishiElectric.co.jp

2nd Tsunato Nakai
Mitsubishi Electric Corporation
Kanagawa, Japan
Nakai.Tsunato@dy.MitsubishiElectric.co.jp

***Abstract*—While the utilization of large language models (LLMs) is expected in various industries, concerns have been raised regarding information leakage and privacy issues related to training data. The application of differential privacy has been actively researched as a solution to the problem. Differential privacy has been applied during the fine-tuning stage of pre-trained LLMs, and the information leakage risk concerning training data has been evaluated through membership inference attacks (MIAs). Although parameter-efficient fine-tuning (PEFT) has recently gained attention, it has also been studied for the application of differential privacy. An evaluation of information leakage risk through MIAs has not yet been reported. Therefore, in this paper, we focused on the representative PEFT method Low-Rank Adaptation(LoRA) , and performed an MIA evaluation of a model trained with differential privacy-enhanced LoRA. We demonstrate the effectiveness of applying differential privacy to LoRA through MIA evaluations. Through the experiments, it was revealed that fine-tuning via LoRA has an effect in mitigating privacy risks. However, no distinct impact on privacy risk reduction was observed through the application of differential privacy. This was consistent even when varying the privacy parameter ϵ in the validation.**

***Index Terms*—Large Language Model, PEFT, Differential Privacy, Membership Inference Attack**

I. Introduction

Transformer-based large language models (LLMs), such as BERT and GPT [1] have been developed and have achieved state-of-the-art performance in various natural language processing tasks. Generally, such models are pre-trained on large and diverse publicly available datasets and then fine-tuned for specific tasks. Fine-tuning is a technique in which a pre-trained model is retrained on a new dataset to improve the task performance. Recently, as the models become larger, parameter-efficient fine-tuning (PEFT) has gained attention. PEFT is a technique used to customize the LLM, where a small number of parameters or layers are added to the original LLM and trained with case-specific data. The weights of the original LLM remain fixed, resulting in significantly fewer parameters being updated during training. This improves performance in specific cases with limited computational resources.

LLMs have a high capacity to memorize training data, leading to concerns regarding the information leakage risks associated with training data [2]. Recently, membership inference attacks (MIAs) have been regarded as an information leakage evaluation tool that reveal the extent to which a model memorizes individual samples from a training dataset and the associated risks.

Ensuring differential privacy (DP) in the training of LLMs has gained attention as a countermeasure to address the information leakage risks associated with training data. DP is a powerful theory in data privacy and a technology that was originally proposed for statistical data. Recently, it has been commonly employed when applying privacy guarantees to machine learning models. Research is underway to apply DP to the training methods of pre-training, fine-tuning, and PEFT in LLMs.

MIA evaluations have begun to be performed on large-scale language models that apply differential privacy, but PEFT has not been evaluated. Du et al. conducted information leakage risk evaluations of training data using MIAs on full fine-tuned models with DP [3]. However, models that are fine-tuned by DP-enhanced PEFT have not been evaluated.

In this study, we investigate whether applying DP to representative PEFT, LoRA can mitigate information leakage risks associated with training data. In the evaluation experiments, we refer to the experiments of Yu et al. [4] and use the LLM GPT-2 and two datasets. We fine-tune the models using DP-enhanced LoRA. Subsequently, we conduct MIA evaluations on the fine-tuned models. Through these experiments, we aim to determine whether applying DP to LoRA method could effectively reduce the information leakage risks associated with the training data.

Contributions. We conduct MIA evaluations on DP-enhanced LoRA, which has not been previously explored. In the experiments, we utilize the GPT-2 which is an LLM for sentence generation and the two datasets; E2E and WebNLG Challenge 2017. We evaluate models fine-tuned with DP-enhanced LoRA. The experimental results reveal that fine-tuning via LoRA has an effect in reducing privacy risks. However, no distinct impact on privacy risk reduction was observed through the application of DP. This was consistent even when we set the privacy parameter ϵ to 0.1, 1, 3, 6 in the validation.

II. Related Work

A case of applying DP to PEFT was reported. Yu et al. reported the accuracy of models fine-tuned with DP using PEFT [4]. Although the accuracy of models fine-tuned with

DP using PEFT has been discussed, the information leakage risks associated with the training data have not been evaluated. Du et al. conducted an information leakage risk evaluation of training data using the MIA for models that trained via DP-enhanced full fine-tuning [3]. To the best of our knowledge, there has been no information leakage risk evaluation of training data for fine-tuning models using PEFT with DP.

III. PRELIMINARIES

In this section, we provide an overview of LoRA, DP, and information leakage related to the training data.

A. LoRA

Hu et al. proposed LoRA which is one of the PEFT method [5]. In our study, we implemented LoRA with DP using the method followed Yu et al. [4]. Within each pre-trained model, there exists a size of dense weight matrix, to which a low-rank matrix is added. Typically, small value is chosen as a rank. This is because most of the parameters in the Transformer architecture are dense weight matrices, and selecting a small rank can dramatically reduce the number of parameters. Hu et al. applied this reparameterization only to the attention weights that constitute the Transformer, while the weights in other feed-forward layers remained fixed.

B. Differential Privacy (DP)

DP is a privacy-protection metric based on indistinguishability [6]. Data which satisfy the DP criterion will preserve security even in the event of new attack methodologies being discovered post data creation, or the emergence of attackers possessing unforeseen background knowledge.

DP is defined as follow.

$$\Pr[M(D) \in O] \leq e^{\varepsilon} \Pr[M(D') \in O] + \delta. \tag{1}$$

When a randomization function $M : D \rightarrow R$ satisfies the above formula for any adjacent databases D and D', M satisfies ϵ-δ DP. O represents any subspace of the output space R. The privacy parameter ϵ represents the maximum distance between the responses when resemble databases the same query. δ represents the probability of unintentional information leakage.

C. Membership Inference Attacks (MIAs)

MIAs determine whether a specific sample is used to train a target model. Recently, it has also been used as an evaluation metric for information leakage risks associated with the training data of machine learning models. The MIA method used in this paper is proposed by Carlini et al [2]. We utilize the percentage of training samples correctly classified as members of the training set as an indicator of privacy risk. Next is how to operate MIA. Samples are fed into both a fine-tuned model and a reference model, and a likelihood ratio is calculated. If this ratio is less than a predetermined threshold, the sample is classified as a member of the training set. This threshold is set such that 10% of validation samples are mistakenly classified as members.

TABLE I
PERPLEXITY AND MIA RECALL OF MODELS TRAINED ONLY PRE-TRAINING, FULL FINE-TUNING AND LORA USING E2E DATASET.

Model Size	method	Perplexity	MIA Recall
GPT-2-Small	Pre-trained	124	0.00
	Full-FT	2.14	1.00
	LoRA	3.65	0.167

IV. EXPERIMENT

In this section, we present the datasets, model architecture, training method, and MIA evaluation.

A. Setup

In this study, experiments were conducted from two perspectives. One is the perspective of the dataset, and the other is the perspective of the model size. Two datasets from natural language generation tasks, WebNLG Challenge 2017 and E2E are used. The WebNLG Challenge 2017 dataset is composed of 18,025 training samples, 2,258 validation samples, and 4,928 test samples. The E2E dataset is composed of 42,061 training samples, 4,672 validation samples, and 4,693 test samples. The models employed in the experiments are GPT-2-Small and GPT-2-Medium. These models excel in language generation and differ in size. GPT-2-Small comprises 12 layers with 110 million parameters, while GPT-2-Medium comprises 24 layers with 336 million parameters. Currently, no established method exists for making appropriate decisions regarding the privacy parameter ϵ settings for mitigating privacy risks when training LLMs. Thus a broad range was set and experiments were conducted. The settings for the privacy parameter ϵ were established as 0.1, 1, 3, and 6. The parameter δ was set as $1e-5$. The dimension of the matrix added by LoRA was set to 4. By using above contents trained for 20 epochs using LoRA with DP, and privacy risk was evaluated using MIAs. We use the MIAs based on the likelihood ratio described in the previous section. The indicators used for evaluation were perplexity, five scores, and MIA Recall. Perplexity is a measure of how accurately a language model can predict the next word. But it does not fully capture the performance of a language model. The score represents the performance of the model based on each method. MIA Recall indicates how successful the attack is. In this research it is used as an index to evaluate how much training data has been leaked.

B. Results And Discussion

Table 1 shows the results of perplexity and MIA Recall for GPT-2-Small that underwent only pre-training, the model that used E2E for full fine-tuning, and the model fine-tuned via LoRA. Both full fine-tuning and fine-tuning via LoRA demonstrated a significant reduction in perplexity, indicating successful training progression. Regarding MIA Recall, the pre-trained model observed 0.00, and full fine-tuning observed 1.00, suggesting that MIA was appropriately implemented. The model fine-tuned via LoRA saw a decrease in MIA Recall

TABLE II
SCORES AND MIA RECALL OF TWO MODELS, FINE-TUNED VIA DP-ENHANCED LORA USING WEBNLG CHALLENGE 2017 AND E2E DATASET ON GPT-2-SMALL AND GPT-2-MEDIUM."W/O DP" INDICATES THE ABSENCE OF DP APPLICATION.

				Score						
	Model Size	ϵ	Perplexity	BLEU	NIST	MET	ROUGE-L	CIDE-r	MIA Recall
WebNLG Challenge 2017	GPT-2-Small	w/o DP	37.9	49.2	9.74	38.1	63.18	3.24	0.124
		0.1	16.5	1.08	1.07	19.4	41.3	1.04	0.137
		1	28.6	20.1	1.26	21.8	45.1	1.24	0.162
		3	29.9	25.9	2.79	25.1	48.8	1.47	0.138
		6	28.0	29.5	4.05	26.8	51.0	1.69	0.154
E2E	GPT-2-Small	w/o DP	3.65	69.6	8.78	46.4	70.9	2.49	0.167
		0.1	7.01	40.9	1.56	30.4	30.4	0.93	0.166
		1	5.77	57.2	5.01	36.1	62.9	1.46	0.155
		3	5.48	53.9	6.40	33.9	59.8	1.56	0.157
		6	5.38	60.6	6.69	37.6	64.6	1.69	0.156
	GPT-2-Medium	w/o DP	3.47	70.2	8.91	46.4	71.2	2.46	0.173
		0.1	9.33	25.0	0.250	23.6	50.9	0.568	0.167
		1	6.60	51.8	4.18	33.6	60.6	1.28	0.157
		3	5.82	56.5	6.21	35.1	62.2	1.56	0.162
		6	5.59	59.1	5.83	36.3	64.6	1.56	0.149

to 0.167. This suggests that LoRA has the effect of reducing privacy risks related to training data. A possible reason is that the number of parameters altered during fine-tuning using LoRA is minimal.

Table 2 presents the experimental results using WebNLG Challenge 2017 and E2E datasets fine-tuned via LoRA with DP. When not applied DP (w/o DP), the model scores are maximized in all categories. Even when utilizing the WebNLG Challenge dataset on GPT-2-Small, the model's score has been observed to decline due to the application of DP. No discernible decreasing trend was identified in relation to MIA Recall due to the application of DP.

Next, we analyze the results using the E2E dataset on GPT-2-Small. When not applied DP (w/o DP), the model scores are maximized in all categories. Furthermore, when altering the privacy parameter ϵ, which regulates the intensity of DP, the model score increases as ϵ becomes larger in all score-related categories, with the lowest score when ϵ is smallest.

A experiment was conducted on models of different sizes. Similarly for GPT-2-Medium, as ϵ larger, the model score also improved. Regarding MIA Recall, 0.012 decrease can be observed with the application of DP, but no significant change in MIA Recall due to DP was detected in GPT-2-Small. In GPT-2-Medium, the application of DP resulted in the MIA Recall decreased by a maximum of 0.023. Upon comparison between GPT-2-Small and GPT-2-Medium, it is discernible that the model with a greater number of parameters, GPT-2-Medium, yields a higher score. Moreover, the degree of degradation due to DP noise is also amplified. In terms of MIA Recall, the proportion of decline due to DP is escalating. Furthermore, in the case of fine-tuning with LoRA, the larger the model size, the greater the effect of applying DP on the risk of information leakage regarding LLM training data by comparing with and without DP. This can be attributed to the increase in the number of parameters to be modified as the model size enlarges.

V. CONCLUSION AND FUTURE WORKS

From the results, it has been elucidated that when fine-tuned with LoRA, it becomes challenging to achieve the effect of reducing privacy risks related to the training data through the application of DP.

In this experiment, we utilized the GPT-2 model. However, it is necessary to conduct similar validations with different models, such as BERT, which excels in language understanding. Additionally, while we evaluated the LoRA method, it is also necessary to apply DP to other PEFT methods and assess their privacy risks.

DP has a trade-off with model accuracy, presenting the disadvantage of performance degradation. Given the results of this study, it appears that the protective effects do not justify the degradation of the model, suggesting the need to consider alternative countermeasures distinct from DP.

REFERENCES

[1] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, "Improving language understanding by generative pre-training,"" OpenAI Report,2018.

[2] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F, "Membership inference attacks from first principles,"" arXiv preprint arXiv:2112.03570, 2021a

[3] Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, Huan Sun, "DP-Forward: Fine-tuning and Inference on Language Models with Differential Privacy in Forward Pass," arXiv preprint arXiv:2309.06746, 2023.

[4] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang, "Differentially Private Fine-tuning of Language Models," arXiv:2110.06500, 2022.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen, "Lora: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis", TCC. 265–284, 2006.

Poster: Towards Extensible Memory Isolation for Low-End Microcontrollers

Marton Bognar
DistriNet, KU Leuven
3001 Leuven, Belgium

Jo Van Bulck
DistriNet, KU Leuven
3001 Leuven, Belgium

***Abstract*—In today's interconnected world, specialized memory isolation mechanisms for low-end embedded microcontrollers are becoming vital. Responding to this need, Texas Instruments has developed the promising Intellectual Property Encapsulation (IPE) feature for their popular ultra-low-power MSP430 microcontrollers. Academic researchers have, furthermore, implemented many trusted execution prototypes on the related openMSP430 open-source softcore. Recent research has shown, however, that these diverging code bases suffer from critical hardware and software vulnerabilities that we believe could have been avoided with more coordination. Hence, we present our work in progress on an openly accessible and extensible research platform based on the specification of the proprietary IPE feature. We implement mature hardware support for IPE on openMSP430, extending it with a fully configurable firmware layer, striving for extensible open-source building blocks that provide a solid foundation for future research.**

1. Introduction

In contrast to high-end servers and desktops, specialized memory isolation features for ubiquitous ultra-low-end embedded devices are generally not as widespread and more limited in functionality. Specifically, some vendors are shipping limited hardware features for code protection and, in certain cases, data isolation in selected microcontrollers. At the same time, a long line of academic research prototypes has explored specialized isolation mechanisms for trusted execution environments (TEEs) specifically aimed at low-end microcontrollers.

A key driver behind these academic innovations has been the availability of open-source softcore implementations, facilitating the rapid prototyping of hardware-software co-designs. OpenMSP430 [8], an open-source reimplementation of the popular line of ultra-low-power MSP430 microcontrollers from Texas Instruments (TI), is such a platform. While openMSP430 itself lacks any built-in security features, it has served as the basis for numerous academic TEE prototypes [6], [7], [12], [13]. Interestingly, newer MSP430 devices produced by TI ship with advanced security features that are not implemented by openMSP430, including a memory protection unit and Intellectual Property Encapsulation (IPE) [18]. IPE utilizes a capable program-counter-based hardware memory isolation mechanism to establish a secluded *enclave* memory region for confidential code and data. This approach has drawn explicit comparisons to the isolation guarantees offered by TEEs [3], and IPE itself has been used as a building block of academic proposals on secure intermittent computing [10], [11].

The emergence and popularity of these security features point to the importance of including hardware memory isolation mechanisms in low-end microcontrollers. Unfortunately, however, current systems have been shown to suffer from critical vulnerabilities, including design flaws and implementation errors [4], improper input sanitization [4], [19], and microarchitectural side channels [4], [20]. Even TI's IPE has been shown vulnerable to different code and data extraction attacks [3], [14], [15], ultimately compromising the security of systems building on it.

Crucially, we observe a large overlap in the impacted security features and hypothesize that many attacks could have been avoided with more coordination. For example, the most crucial vulnerability on IPE, controlled `call` corruption [3], assigned high severity by TI, was found to be already mitigated via a minimal hardware change in both Sancus and VRASED, two research systems built on openMSP430. Similarly, IPE does not enforce a single entry point, a well-understood design requirement for embedded TEEs [7], [9], [12], [13], [16]. IPE has, furthermore, been shown to straightforwardly leak register contents on interrupts [3], [14], which is avoided in Sancus and VRASED via custom secure interrupt extensions or guaranteed atomic execution [1], [5], [6], [13]. Conversely, VRASED's direct memory access (DMA) access control check was shown to suffer from a subtle implementation bug that is not present in either Sancus or IPE [4]. Given these remarkable overlaps, we can only assume that other closed-source (openMSP430-based) implementations likely suffer from similar recurring vulnerabilities.

Our vision. In general, we observe that there is a niche for low-end embedded TEE research prototypes, which often share remarkably similar base functionality. However, the current practice is to reimplement this functionality, commonly introducing (known) vulnerabilities or overlooking pitfalls. Moreover, ongoing efforts on vetting [4], [19] and formal verification [5], [13] are necessarily specific to particular prototype implementations. We hypothesize that having a common base implementation of an isolation feature that can easily be extended to provide more complex security guarantees is beneficial for the community.

Our contributions. We propose an extensible hardware-software co-design that closely mimics and extends the base specification of TI's proprietary IPE, including a fully configurable trusted firmware layer that is transparently executed on reset and remains immutable during program

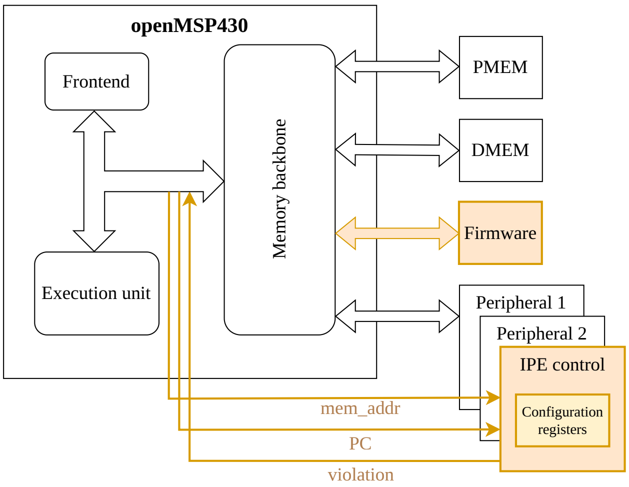

Figure 1. Simplified schematic overview of our design extending the openMSP430 core. Two new components are added: *firmware* memory and *IPE control*, the peripheral implementing the access control logic based on the highlighted added connections.

execution. Our envisioned unified framework will provide strong and flexible building blocks for rapid prototyping of novel embedded memory isolation security mechanisms. Importantly, an eventual open-source implementation means that efforts will not have to be duplicated, and improvements, tests, and verification efforts of the common parts will be reusable in all derived designs.

Furthermore, we plan to streamline software development via a tailored open-source toolchain providing source-level compatibility with current IPE projects. This will also enable research on compiler modifications [21], currently restricted by TI's proprietary IPE compiler.

2. Design and implementation

Like numerous academic prototypes [2], [6], [7], [12], [13] discussed above, we implement our design on the open-source openMSP430 [8] softcore. Figure 1 overviews the main components and connections we added to the base openMSP430 architecture. The main changes (highlighted in orange) are a new *IPE control* peripheral that implements the access control logic and a *firmware* layer that is securely executed on device reset.

Threat model. For our design, we start from the threat model of IPE, designed to protect intellectual property on a device that is controlled by an untrusted party. In this original model, the attacker controls external interfaces such as the DMA controller and the JTAG debug unit, and all untrusted software on the device except the IPE region and a small firmware layer (*bootcode*). In addition, the FRAM technology used in these devices offers certain protections [17] against attackers with physical access.

With our implementation, we try to stay as close as possible to this model. Particularly, we consider *all* untrusted software, the DMA controller, and the openMSP430 debug unit to be compromised. As our prototype implementation is not a physical design, protecting against physical attacks on the memory is outside our scope. It is, however, possible to implement our design on an FPGA and connect an external memory chip with the same properties as TI's FRAM. Likewise, similar to IPE, we rely on the correct functioning of a small trusted firmware layer, which can also be implemented as an external memory unit, in which case adding physical protections to this external chip may be appropriate.

TABLE 1. ACCESS CONTROL RIGHTS. PERIPHERALS AND DMEM ARE OMITTED FROM THE ROWS, AS THEY CANNOT EXECUTE CODE. IPE ENTRY IS AN EXTENSION OF THE ORIGINAL SPECIFICATION [3].

	Peripherals	DMEM	Firmware	PMEM	IPE	IPE entry
Firmware	`rw-`	`rw-`	`rwx`	`rwx`	`---`	`--x`
IPE + entry	`rw-`	`rw-`	`r--`	`rwx`	`rwx`	`rwx`
PMEM	`rw-`	`rw-`	`r--`	`rwx`	`---`	`--x`
DMA	`rw-`	`rw-`	`r--`	`rw-`	`---`	`---`
Debug unit	`rw-`	`rw-`	`r--`	`rw-`	`---`	`---`

Address space. In openMSP430, the 16-bit memory address space is divided into three partitions: read-only program memory (PMEM), non-executable data memory (DMEM), and memory-mapped peripherals. The size of the memory partitions is configurable, and they are connected with separate buses to the memory backbone handling the arbitration (cf. Figure 1). We base our implementation on the MSP430FR5969, which places the IPE configuration registers at base address `05A0h`, making it necessary to extend the openMSP430 peripheral space to 4 kB. Similarly, to better align with the writeable and executable FRAM memory in TI devices, we modified the memory backbone and two-stage pipeline to support write operations to program memory, enabling dynamic data updates for the secluded IPE region inside PMEM.

Memory access control. An important aspect of our system is the implementation of memory access control based on the IPE specification. The IPE region, determined by the value of two boundary configuration registers, is protected from outside software, DMA, and debugger accesses, as summarized in Table 1. On TI devices featuring a 20-bit address space, these configuration registers store the most significant 10 bits of the boundaries, aligning the region to 1 kB boundaries. In our implementation, we adhere to this design for compatibility reasons, but as openMSP430 only has a 16-bit address space, storing full boundary addresses and implementing byte-granular protection boundaries is a feasible alternative.

The access control is handled largely by the IPE control peripheral (cf. Figure 1), which stores the configuration registers and is connected to the CPU by several monitoring and control signals, signaling access violations to the memory backbone. Illegal reads immediately return the value `3FFF` without forwarding the operation to the memory, preventing possible microarchitectural leakage by the resulting value's propagation. Illegal stores are also suppressed, while illegal jumps trigger a non-maskable interrupt. The access control checks are also performed for accesses by the debug unit and DMA requests. Moreover, during the execution of IPE, the debug unit is disabled, preventing it from leaking or corrupting register values.

Secure firmware. In TI devices, special *bootcode* is responsible for setting up the IPE protection before untrusted code can execute or the debugger can attach. While TI did not release its source code and it is not stored in an accessible memory location on the microcontrollers, we reimplemented this bootcode as a modifiable firmware layer based on TI documentation. We will open-source the bootcode and make it readable from software on the device, as we believe that availability leads to more scrutiny and better security down the line. When implementing extensions to our system, open access also allows the extension and modification of this firmware layer, possibly implementing more complex functionality leading to different security guarantees.

Interestingly, this firmware memory needs to enjoy similar protections as the IPE region itself (cf. Table 1), as it needs to be protected from tampering by the attacker, which could undermine the security of the system, such as the correct setup of the IPE region. We implemented these protections in an analogous way to the IPE access rights explained earlier.

3. Ongoing and future work

Hardware extensions. IPE Exposure [3] proposed minimal hardware extensions to IPE-enabled microcontrollers to mitigate the issues discovered in their analysis. These fixes enforce a single entry point to the IPE region and eliminate the controlled `call` corruption vulnerability. We have already implemented these hardware fixes in our prototype, and we will evaluate their cost and performance as compared to software workaround fixes [3].

Additional efforts could focus on extending IPE's access control policies, e.g., by supporting more than one protected region [12], securing I/O devices [7], [12], or interruptible protected execution [1], [5], [6].

Software toolchain and extensions. While we cannot offer full binary compatibility with TI IPE due to its 20-bit address space and extended instructions, we aim to provide full software compatibility of C projects, only requiring recompilation of code written for TI. We base our toolchain on TI's example IPE project [18] and the software mitigation framework introduced in IPE Exposure [3], the latter of which already provides some protection against interface sanitization attacks [19]. At the application level, generic hardware support for secure IPE code and data regions on openMSP430 enable new use cases, e.g., secure intermittent computing [10], [11], real-time guarantees [1], or software-based attestation [13].

Security tests. To systematically validate the correct workings of the IPE bootcode and the memory access control logic, we are developing an extensive unit test framework running in the `iverilog` simulator. Future work may strengthen security guarantees via more complex test cases or even formal verification [5], [13].

Acknowledgements

This research was partially funded by the ORSHIN project (Horizon Europe grant agreement #101070008), the Research Foundation – Flanders (FWO) via grants #1261222N and #G081322N, the Research Fund KU Leuven, and the Cybersecurity Research Program Flanders.

References

[1] Fritz Alder, Jo Van Bulck, Frank Piessens, and Jan Tobias Mühlberg. Aion: Enabling open systems through strong availability guarantees for enclaves. In *CCS*, 2021.

[2] Fatemeh Arkannezhad, Justin Feng, and Nader Sehatbakhsh. Ida: Hybrid attestation with support for interrupts and toctou. In *NDSS*, 2024.

[3] Marton Bognar, Cas Magnus, Frank Piessens, and Jo Van Bulck. Intellectual property exposure: Subverting and securing Intellectual Property Encapsulation in Texas Instruments microcontrollers. In *USENIX Security*, 2024.

[4] Marton Bognar, Jo Van Bulck, and Frank Piessens. Mind the gap: Studying the insecurity of provably secure embedded trusted execution architectures. In *S&P*, 2022.

[5] Matteo Busi, Job Noorman, Jo Van Bulck, Letterio Galletta, Pierpaolo Degano, Jan Tobias Mühlberg, and Frank Piessens. Provably secure isolation for interruptible enclaved execution on small microprocessors. In *CSF*, 2020.

[6] Ruan De Clercq, Frank Piessens, Dries Schellekens, and Ingrid Verbauwhede. Secure interrupts on low-end microcontrollers. In *ASAP*, 2014.

[7] Karim Eldefrawy, Gene Tsudik, Aurélien Francillon, and Daniele Perito. SMART: Secure and minimal architecture for (establishing dynamic) root of trust. In *NDSS*, 2012.

[8] Olivier Girard. openmsp430 rev 1.17. https://github.com/olgirard/openmsp430/blob/master/doc/openMSP430.pdf, 2017.

[9] Patrick Koeberl, Steffen Schulz, Ahmad-Reza Sadeghi, and Vijay Varadharajan. TrustLite: a security architecture for tiny embedded devices. In *EuroSys*, 2014.

[10] Archanaa S Krishnan and Patrick Schaumont. Benchmarking and configuring security levels in intermittent computing. *ACM TECS*, 21(4), 2022.

[11] Archanaa S Krishnan, Charles Suslowicz, and Patrick Schaumont. Secure and stateful power transitions in embedded systems. *Journal of Hardware and Systems Security*, 4, 2020.

[12] Job Noorman, Jo Van Bulck, Jan Tobias Mühlberg, Frank Piessens, Pieter Maene, Bart Preneel, Ingrid Verbauwhede, Johannes Götzfried, Tilo Müller, and Felix C. Freiling. Sancus 2.0: A low-cost security architecture for IoT devices. *ACM Transactions on Privacy and Security*, 20(3), 2017.

[13] Ivan De Oliveira Nunes, Karim Eldefrawy, Norrathep Rattanavipanon, Michael Steiner, and Gene Tsudik. VRASED: A verified hardware/software co-design for remote attestation. In *USENIX Security*, 2019.

[14] Prakhar Sah and Matthew Hicks. RIPencapsulation: Defeating IP encapsulation on TI MSP devices, 2023.

[15] Marc Schink and Johannes Obermaier. Taking a look into execute-only memory. In *WOOT*, 2019.

[16] Raoul Strackx, Frank Piessens, and Bart Preneel. Efficient isolation of trusted subsystems in embedded systems. In *Security and Privacy in Communication Networks*, 2010.

[17] Texas Instruments. Closing the security gap with TI's MSP430 FRAM-based microcontrollers. https://www.ti.com/lit/wp/slay035/slay035.pdf, 2014.

[18] Texas Instruments. MSP code protection features. https://www.ti.com/lit/an/slaa685/slaa685.pdf, 2015.

[19] Jo Van Bulck, David Oswald, Eduard Marin, Abdulla Aldoseri, Flavio D. Garcia, and Frank Piessens. A tale of two worlds: Assessing the vulnerability of enclave shielding runtimes. In *CCS*, 2019.

[20] Jo Van Bulck, Frank Piessens, and Raoul Strackx. Nemesis: Studying microarchitectural timing leaks in rudimentary CPU interrupt logic. In *CCS*, 2018.

[21] Hans Winderix, Jan Tobias Mühlberg, and Frank Piessens. Compiler-assisted hardening of embedded software against interrupt latency side-channel attacks. In *EuroS&P*, 2021.

Poster: Privacy-Preserving Billing for Local Energy Markets

Eman Alqahtani*and Mustafa A. Mustafa*†
*Department of Computer Science, The University of Manchester, Manchester, M13 9PL, UK
‡COSIC, KU Leuven, Leuven, 3001, Belgium
Email: eman.alqahtani@postgrad.manchester.ac.uk, mustafa.mustafa@manchester.ac.uk

***Abstract*—We propose a privacy-preserving billing protocol for local energy markets (PBP-LEMs) that takes into account market participants' energy volume deviations from their bids. It enables a group of market entities to jointly compute participants' bills in a decentralized and privacy-preserving manner. It also mitigates risks on individuals' privacy arising from any potential internal collusion. We first propose a novel, efficient, and privacy-preserving individual billing scheme, achieving information-theoretic security, which serves as a building block. PBP-LEMs utilises this scheme, along with other techniques such as multiparty computation, Pedersen commitments and inner product functional encryption, to ensure data confidentiality and accuracy. We present three approaches, resulting in different levels of privacy and performance. We also show that PBP-LEMs is feasible for deployment in real LEMs.**

***Index Terms*—Security, Privacy, LEM, MPC**

I. Introduction

Local energy markets (LEMs) enable prosumers – individuals who consume and produce – to actively participate in open markets, supporting direct trading of surplus energy with others. This contrasts with the traditional practice of selling to suppliers at feed-in-tariff (FiT), which is lower than retail market price (RP).

LEMs typically require participants to submit bids before the actual trading period (e.g., 1 hour in advance). Therefore, market participants need to forecast their bid volumes, indicating the amount of energy they intend to trade. This prediction relies on historical data and estimated consumption, making it inherently prone to errors. As a result, participants may commit to trade specific energy volumes but subsequently fail to fulfill these commitments. As this failure can disrupt grid stability and increase costs, market participants should be accounted for their deviations from their committed bid volumes during the billing process. Furthermore, a market participant's deviation may cause a different level of effect on the grid depending on what part of the grid the participant resides in. As a result, some might bear higher costs than others, not only due to their deviation amounts but also because of their specific locations on the grid.

The computation and settlement of bills/rewards for participants in LEMs, while considering their deviations, requires critical private information, including individual bid volumes and meter readings per trading period. These are closely associated with individuals' consumption patterns, posing risks to their privacy.

Previous studies have addressed the privacy concerns during billing and settlement, employing techniques such as anonymization, perturbation , and homomorphic encryption. Nevertheless, only a few have considered establishing a billing mechanism based on the actual amount of energy produced/consumed by LEM participants [1, 2] or to additionally account for deviations [3–5]. However, the billing process is coordinated or executed by a single honest party [1, 2, 4], disclosing some or all of the individuals private data to the party and potentially imposing a high computational cost on a single party. This raise questions about the applicability or scalability of their solutions in realistic scenarios. In contrast, solutions [3, 5] rely on a group of entities working together to compute individual bills/rewards privately. However, they are vulnerable to the risk of internal collusion, potentially leading to the disclosure of highly sensitive data.

To address these limitations, we propose a privacy-preserving billing protocol for LEMs (PBP-LEMs), considering participants' deviations. Our contributions are:

- We propose a novel efficient and privacy-preserving individual billing scheme (EPIBS).
- We design a privacy-preserving billing protocol (PBP-LEMs) considering participants' energy volume deviations and their locations on the grid, by utilising EPIBS, Pedersen commitments, multiparty computation (MPC), and function-hiding inner product encryption (FHIPE). PBP-LEMs involves a collaboration of different entities performing bill computation without revealing any individual participant's private data. It also mitigates the potential impact on individuals' privacy resulting from internal collusion. We propose three approaches, resulting in different levels of privacy and performance.
- We evaluate the computation and communication complexity of PBP-LEMs.

II. Preliminaries

System Model (Fig. 1): Users, supported by smart meters (SMs), engage in a LEM by submitting bids (consisting of energy volume and offered price). *Local Energy Market Operator (LEMO)* executes the LEM, identifying the trading price and the accepted bids. *Suppliers* provide energy to their customers in the retail market at RP and purchase electricity injected by their customers that is not traded in the LEM at FiT. They also generate and manage their customers' (i.e., users) bills based on their participation in the LEM. *Distribution System Operator (DSO)* divides the LEM area into

This work was supported by EPSRC through EnnCore [EP/T026995/1] and by the Flemish Government through FWO-SBO SNIPPET [S007619].

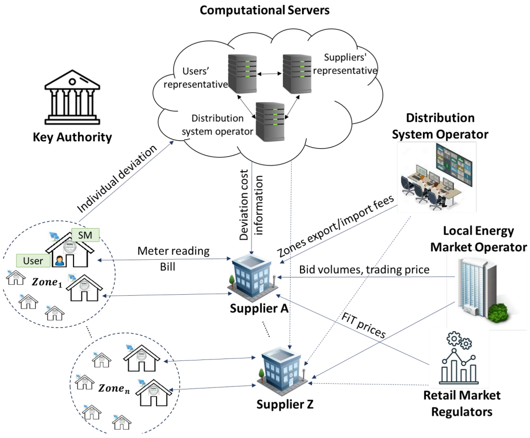

Fig. 1: System model.

TABLE I: Notations

Symbol	Description
bp_u, tp_k	u-th billing period, k-th trading period, $k \in \{1, \ldots, N_k\}$.
Id_i	Unique identifier of user i, $i \in \{1, \ldots, N_u\}$.
SId_j, ZId_z	Unique identifier of supplier j, zone z.
$b_i^{tp_k}, m_i^{tp_k}$, $d_i^{tp_k}$, $s_i^{tp_k}$	Bid volume, meter reading, type of participation (binary), deviation cost inclusion state (binary, i.e., 1 if the user has to pay a deviation cost and 0 otherwise) of user i at tp_k.
$C_{i,p}^{tp_k}$, $C_{i,c}^{tp_k}$	Condition that requires user i to pay/get paid to/by supplier due to his/her deviation at tp_k.
$dev_p^{tp_k}/dev_c^{tp_k}$	cost/compensation to be paid/received by certain prosumers/consumers to/from their suppliers due to deviations based on a condition, $C_{i,p}^{tp_k}$p/$C_{i,c}^{tp_k}$.
T^{tp_k}, W^{tp_k}	Total deviation, zonal deviation weight at tp_k.
$t_z^{tp_k}, np_z^{tp_k}, nc_z^{tp_k}$	Total deviation, number of prosumers, number of consumers in zone z at tp_k.
$NF_p^{z,tp_k}/NF_c^{z,tp_k}$	Network fee for exporting/importing in zone z at tp_k.

Algorithm 1 EPIBS

```
• $EPIB.Setup()$
    1) Generate three sets of encryption keys $S_i = \{sk_i^{tp_1}, \ldots, sk_i^{tp_{N_k}}\}$, $S_i^p = \{sk_{i,p}^{tp_1}, \ldots, sk_{i,p}^{tp_{N_k}}\}$ and $S_i^c = \{sk_{i,c}^{tp_1}, \ldots, sk_{i,p}^{tp_{N_k}}\}$
• $EPIB.MeterEncrypt(m_i^{tp_k})$
    1) Compute $mc_i^{tp_k} = m_i^{tp_k} + sk_i^{tp_k}$.
• $EPIB.TypeEncrypt(d_i^{tp_k})$
    1) Encode $d_i^{tp_k}$ into $d_{i,p}^{tp_k}$ and $d_{i,c}^{tp_k}$, each representing a binary state of the user's type participation at $tp_k$ (e.g., if the user is buying energy, then $d_{i,p}^{tp_k} = 1$ and $d_{i,c}^{tp_k} = 0$).
    2) Compute $dc_{i,p}^{tp_k} = d_{i,p}^{tp_k} + sk_{i,p}^{tp_k}$ and $dc_{i,c}^{tp_k} = d_{i,c}^{tp_k} + sk_{i,c}^{tp_k}$.
• $EPIB.BillCompute(mc_i^{tp_k}, dc_{i,p}^{tp_k}, dc_{i,c}^{tp_k})$
    1) Compute $bc_i^{tp_k} = mc_i^{tp_k} * (TP^{tp_k} + (dc_{i,p}^{tp_k} * NF_p^{tp_k}) + (dc_{i,c}^{tp_k} * NF_c^{tp_k})) + [C_{i,p}^{tp_k}](dc_{i,p}^{tp_k} * dev_p^{tp_k}) + [C_{i,c}^{tp_k}](dc_{i,c}^{tp_k} * dev_c^{tp_k})$
    2) $BLc_i^{bp_u} = \sum_{k=1}^{N_k} bc_i^{tp_k}$
• $EPIB.DecKeyGen(S_i,\ S_i^c,\ S_i^p, )$
    1) Compute $dk_i^{dev,tp_k} = [C_{i,p}^{tp_k}](sk_{i,p}^{tp_k} * dev_p^{tp_k}) + [C_{i,c}^{tp_k}](sk_{i,c}^{tp_k} * dev_c^{tp_k})$
    2) Compute $dk_i^{tp_k} = (sk_i^{tp_k} * TP^{tp_k}) + (sk_{i,p}^{tp_k} * NF_p^{tp_k}) + (sk_{i,c}^{tp_k} * NF_p^{tp_k}) + dk_i^{dev,tp_k}$
    3) Compute $DK_i^{bp_u} = \sum_{k=1}^{N_k} dk_i^{tp_k}$
• $EPIB.Decrypt(DK_i^{bp_u}, BLc_i^{bp_u})$
    1) Compute $DK_i^{bp_u} = \sum_{k=1}^{N_k} dk_i^{tp_k}$
```

zones and sets energy importing/exporting fees for each zone. *Computational Severs (CS)* calculate the required deviation cost information according to the zone-based billing model with universal cost split (ZBUCS) [5]. *Key Authority (KA)* generates the keys.

Threat Model: LEMO, RMRs and suppliers are semi-honest. Users and external entities are malicious. We assume two security settings for CS: honest majority (HM) and dishonest majority (DM) with active adversaries. We assume that the communication channels are private and authentic, entities are time-synchronized and SMs are tamper-proof.

Functional Requirements: (i) Each supplier learn each of their customers' aggregated bills/rewards generated from their participation in LEM per billing period (ii) Each supplier learn its income balance incurred from selling (buying) the deviation shares to (from) their customers per trading slot (iii) Each user learn their own LEM bill/reward per billing period.

Security and Privacy Requirements: (i) Privacy preservation: exact users' locations and their type of participation (selling or buying) should be hidden. (ii) Confidentiality: users' bid volumes, meter readings, deviations, and bills/rewards per trading period from LEM should be hidden. (iii) Collusion Impact Mitigation: users' highly sensitive data should not be revealed in the event of internal collusion.

Building Blocks: MPC PBP-LEMS utilises the following building blocks: (i) MPC protocols proposed by [6] and [7] for honest-majority dishonest-majority settings, respectively; (ii) Pedersen Commitment [8]; (iii) FHIPE [9] (details below);

- $IPE.Setup(1_\lambda, S) \rightarrow (pp, msk)$ given a security parameter λ, Setup initializes the system and outputs the public parameters pp and master key msk.
- $IPE.LeftEncrypt(msk, \alpha, x) \rightarrow Ct_x$ given the master key msk, a vector x and a uniformly random element α, LeftEncrypt outputs a ciphertext Ct_x.
- $IPE.RightEncrypt(msk, \beta, y) \rightarrow Ct_y$ given the master key msk, a vector y and a uniformly distributed element β, RightEncrypt outputs a ciphertext Ct_y.
- $IPE.Decrypt(pp, Ct_x, Ct_y) \rightarrow z$ given pp, ciphertexts Ct_x and Ct_y, Decrypt outputs z such that $z =< x, y >$.

and (iv) Dual binary encoding scheme for integer comparison using inner products [2] which converts two numbers, x and y, into arrays of vectors to enable comparing them by means of multiple inner products:

- $f_x(x) \rightarrow (X_{v_l}, X_{v_g})$: on input a number x, f_x output two encoded arrays of vectors X_{v_l} and X_{v_g}.
- $f_y(y) \rightarrow Y_v$: on input a number y, f_y output one encoded array of vectors Y_v.

III. PBP-LEMs

Operating over a finite field modulo $q(z_q)$, EPIBS consists of the methods shown in Algorithm 1. PBP-LEMs consists of four steps shown in Algorithm 2. $[x]$ denotes a secret sharing of x and $<x>$ represents a Pedersen commitment to x.

IV. Performance Evaluation

Suppliers and CS were implemented on a Linux server with 16-core Intel Xeon and 64 GB of memory. Tests were preformed on realistic dataset. Bills computation in all approaches, can be completed maximally in less than 5 minutes

Algorithm 2 Privacy-Preserving Billing Protocol

. . . . Prerequisites . . . .

1) User sends a registration request along with (Id_i, SId_j, ZId_z) to KA.
2) KA generates $(S_i, S_i^c, S_i^p) \leftarrow EPIBS.Setup()$ and $(pp_i, msk_i) \leftarrow IPE.Setup(1_\lambda, S)$; and sends $(S_i,\ S_i^c,\ S_i^p, pp_i, msk_i)$ the user.
3) We assume that users have performed the following before the market execution $\langle -b_i^{tp_k} \rangle \leftarrow Commit(-b_i^{tp_k}, r_{i,b}^{tp_k})$, $(dc_{i,p}^{tp_k}, dc_{i,c}^{tp_k}) \leftarrow EPIBS.TypeEncrypt(d_i^{tp_k})$, $(B_{i,j,vl}^{tp_k}, B_{i,vg}^{tp_k}) \leftarrow f_x(b_i^{tp_k})$, $Bc_{i,vl}^{tp_k} \leftarrow (IPE.RightEncrypt(msk_i, \beta, B_{i,j,vl}^{tp_k})$ for all vectors of $Bc_{i,vl}^{tp_k})$, $Bc_{i,vg}^{tp_k} \leftarrow (IPE.RightEncrypt(msk_i, \beta, B_{i,j,vg}^{tp_k})$ for all vectors of $Bc_{i,vg}^{tp_k})$ and sent $(Id_i\langle -b_i^{tp_k}\rangle, dc_{i,p}^{tp_k}, dc_{i,c}^{tp_k} Bc_{i,v_l}^{tp_k}, B_{i,v_l}^{tp_k})$ to LEMO.

. . . . . Step 1: Input Data Generation and Distribution Per tp_k . . . . . .

1) Every user calculates their deviation: $v_i^{tp_k} = b_i^{tp_k} - m_i^{tp_k}$ and sends $(Id_i, SId_j, ZId_z, [v]_i^{z,tp_k}, [d]_i^{z,tp_k})$ to CS.
2) Each SM computes $M_i^{tp_k} \leftarrow f_y(m_i^{tp_k})$, $Mc_i^{tp_k} \leftarrow (IPE.LeftEncrypt(msk_i, \alpha, M_{i,j}^{tp_k})$ for all vectors of $M_i^{tp_k})$, $mc_i^{tp_k} \leftarrow EPIBS.MeterEncrypt(m_i^{tp_k})$, $\langle m_i^{tp_k} \rangle \leftarrow Commit(m_i^{tp_k}, r_{i,m}^{tp_k})$ and sends $(Id_i, ZId_z, mc_i^{tp_k}, Mc_i^{tp_k}, \langle m_i^{tp_k} \rangle)$ to the supplier.
3) LEMO forwards $(Id_i, Bc_{i,v_r}^{tp_k}, Bc_{i,v_l}^{tp_k}, \langle -b_i^{tp_k} \rangle, dc_{i,p}^{tp_k}, dc_{i,c}^{tp_k})$ for each user to the supplier.

. . . . Step 2: Bills Computation Per tp_k . . . .

1) CS execute algorithm 3 for each zone using MPC protocols.
2) CS reconstruct $t_z^{tp_k}$, $nc_z^{tp_k}$ and $np_z^{tp_k}$ for each zone, compute (T^{tp_k}, W^{tp_k}) based on ZBUCS [5]; and publish $ZN = (t_z^{tp_k}, nc_z^{tp_k}, np_z^{tp_k})$ tuples and (T^{tp_k}, W^{tp_k}).
3) Determining the deviation cost inclusion state, $s_i^{tp_k}$ for each user, utilising $(t_z^{tp_k}, T^{tp_k})$, using one of three approaches.
   - Approach 1: Each supplier identifies $s_i^{tp_k}$ for their customers in oblivious fashion using the encoding and comparison scheme of [2]; and $IPE.Decrypt()$ over $(Bc_{i,vl}^{tp_k}, Bc_{i,v_g}^{tp_k}, Mc_i^{tp_k})$.
   - Approach 2: CS identify $[s_i^{tp_k}]$ by executing MPC comparison operations on $[v_i^{tp_k}]$.
   - Approach 3: $v_i^{tp_k}$ are disclosed to customers' suppliers.
4) Each supplier computes $bc_i^{tp_k}$ for their customers (Algorithm 4).

. . . . Step 3: Bills Computation and Distribution per bp_u . . . .

1) Each supplier computes $BLc_i^{bp_u} = \sum_{k=1}^{N_k} bc_i^{tp_k}$
2) KA computes $DK_i^{bp_u} \leftarrow EPIBS.DecKeyGen(bc_i^{tp_k}, dc_{i,p}^{tp_k}, dc_{i,c}^{tp_k})$
3) The supplier computes $BL_i^{bp_u} \leftarrow EPIBS.Decrypt(DK_i^{bp_u}, BLc_i^{bp_u})$ and sends it to the user.

. . . . Step 4: Individual Deviations Verification . . . .

1) For every tp_k, the supplier computes $\langle v_i^{tp_k} \rangle = \langle m_i^{tp_k} \rangle . \langle -b_i^{tp_k} \rangle$. At the end of bp_u:
2) The supplier computes $< \sum_{k=1}^{N_k} v_i^{tp_k} >= \prod_{k=1}^{N_k} \langle v_i^{tp_k} \rangle$.
3) CS compute $[V_i^{bp_u}] = \sum_{k=1}^{N_k} [v_i^{tp_k}]$. and send it to the supplier.
4) Each SM computes $R_i^{bp_u} = \sum_{k=1}^{N_k} (r_{i,m}^{tp_k} + r_{i,b}^{tp_k})$.
5) Supplier verifies $Open(< \sum_{k=1}^{N_k} v_i^{tp_k} >, V_i^{bp_u}, R_i^{bp_u}) \rightarrow true$.

for 4000 users, showing the feasibility of our protocol for real LEM deployment (Fig. 3).

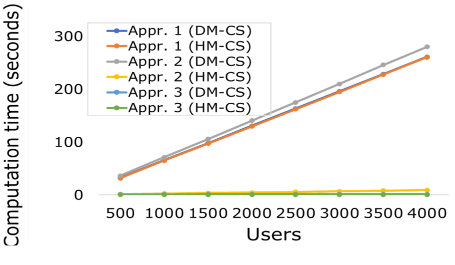

Fig. 2: Total bills computation cost per trading period

Algorithm 3 Zone-based Deviations Aggregation

```
Input: Set of $N_u^z$ user tuples $U = ([v], [d])$ who belong to zone z
Output: Zone $z$ tuple $ZN = ([t], [np], [nc])$
for $i = 0$ to $N_u^z$ do
    $[t]_z^{tp_k} \leftarrow [t]_z^{tp_k} + [v]_i^{tp_k}$
    $[p]_z^{tp_k} \leftarrow [p]_z^{tp_k} + [d]_i^{tp_k}$
    $[c]_z^{tp_k} \leftarrow [p]_z^{tp_k} + 1 - [d]_i^{tp_k}$
end for
```

Algorithm 4 Individual Bill Computation Per Trading Period

```
Input: $mc_i^{tp_k}$, $s_i^{tp_k}$, $dc_{i,p}^{tp_k}$, $dc_{i,c}^{tp_k}$, $(T^{tp_k}, W^{tp_k})$, zone z tuple $ZN = (t_z^{tp_k}, nc_z^{tp_k}, np_z^{tp_k})$ to which the user belongs.
Output: $bc_i^{tp_k}$, $C_{i,p}^{tp_k}$ , $C_i^{c,tp_k}$
$bc_i^{tp_k} \leftarrow mc_i^{tp_k} * (TP^{tp_k} + (dc_{i,p}^{tp_k} \times -NF_p^{z,tp_k}) + (dc_{i,c}^{tp_k} \times NF_c^{z,tp_k}))$
$C_{i,p}^{tp_k} \leftarrow (T^{tp_k} > 0 \textbf{ and } s_i^{tp_k})$
$C_{i,c}^{tp_k} \leftarrow (T^{tp_k} < 0 \textbf{ and } s_i^{tp_k})$
$dev_p^{tp_k} \leftarrow C_{i,p}^{tp_k} \times (t_z^{tp_k} \times W^{tp_k}/np_z^{tp_k}) \times (FiT^{tp_k} - TP^{tp_k})$
$bc_i^{tp_k} \leftarrow bc_i^{tp_k} + dc_{i,p}^{tp_k} \times dev_p^{tp_k}$
$dev_c^{tp_k} \leftarrow C_{i,c}^{tp_k} \times (t_z^{tp_k} \times W^{tp_k}/nc_z^{tp_k}) \times (RP^{tp_k} - TP^{tp_k})$
$bc_i^{tp_k} \leftarrow bc_i^{tp_k} + dc_{i,c}^{tp_k} \times dev_c^{tp_k}$
```

Communication cost varies significantly based on the applied approach as shown in Fig. 2.

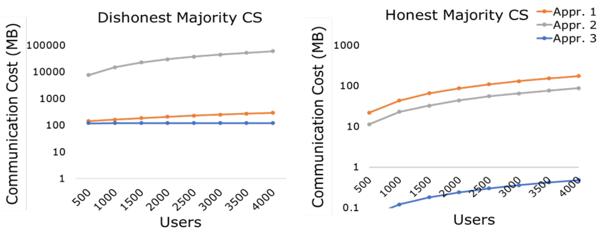

Fig. 3: Total communication cost per trading period

V. Conclusions

We introduced a privacy-preserving billing protocol for LEMs and demonstrated its feasibility in real-world settings.

References

[1] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[2] T. Gaybullaev, H.-Y. Kwon, T. Kim, and M.-K. Lee, "Efficient and privacy-preserving energy trading on blockchain using dual binary encoding for inner product encryption," *Sensors*, vol. 21, no. 6, 2021.

[3] K. Erdayandi, L. C. Cordeiro, and M. A. Mustafa, "A privacy-preserving and accountable billing protocol for peer-to-peer energy trading markets," in *Int. Conf. on Smart Energy Systems and Technologies*, 2023.

[4] A. Hutu and M. A. Mustafa, "Privacy preserving billing in local energy markets with imperfect bid-offer fulfillment," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2023.

[5] E. Alqahtani and M. A. Mustafa, "Zone-based privacy-preserving billing for local energy market based on multiparty computation," 2023.

[6] K. Chida, D. Genkin, K. Hamada, D. Ikarashi, R. Kikuchi, Y. Lindell, and A. Nof, "Fast large-scale honest-majority mpc for malicious adversaries," in *Advances in Cryptology – CRYPTO 2018*. Springer, 2018, pp. 34–64.

[7] M. Keller, E. Orsini, and P. Scholl, "Mascot: Faster malicious arithmetic secure computation with oblivious transfer," in *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, p. 830–842.

[8] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO*, 1992, pp. 129–140.

[9] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical," in *Security and Cryptography for Networks*, D. Catalano and R. De Prisco, Eds. Cham: Springer International Publishing, 2018, pp. 544–562.

Poster:
Towards a Digital Payment System for the Constrained Internet of Things

Mikolai Gütschow
TU Dresden
mikolai.guetschow@tu-dresden.de

Matthias Wählisch
TU Dresden and Barkhausen Institut
m.waehlisch@tu-dresden.de

***Abstract*—In this poster, we start the discussion of the potentials and challenges of digital payment systems to advance digital services in the Internet of Things. We specifically focus on devices with constrained hardware resources. To enable multi-stakeholder machine-to-machine scenarios, we propose an e-cash approach that is privacy-friendly and allows for autonomous payment. We implement our approach using GNU Taler, a free-software e-cash implementation, and RIOT, a free and open-source operating system for the IoT. Our preliminary findings suggest that the deployment of e-cash systems is feasible in constrained IoT scenarios. They underscore the importance of concise, standard-compliant data encoding over computationally intensive compression techniques.**

1. Introduction

The Internet of Things (IoT) is projected to consist of 30 billion interconnected devices by 2030 [1]. Most of them will be constrained in terms of hardware resources to reduce manufacturing costs, enabling mass deployment of many different new applications. In principle, each of these IoT devices provides a service (*e.g.,* sensing data, acting to external input), often in multi-stakeholder environments in which not all stakeholders necessarily collaborate in a peer-to-peer manner. How to seamlessly offer advanced services in such networks is still an open topic.

Providing an economic incentive could be one reason for cooperation. To enable, for example, data sharing between different stakeholders then requires *autonomous machine-to-machine (M2M) payments* such that the payment is integrated with and running directly on the (constrained) IoT devices, keeping the overhead of accounting and payment processing low. Additionally, such an IoT payment system must prioritize *(meta-)data privacy protection* due to the sensitivity and scale of data involved.

In this poster, we propose payments in the IoT based on blind signatures and a centralized architecture. Such a token-based approach allows for payer privacy and enables autonomous usage by design, thereby meeting two fundamental requirements. Section 2 gives some examples of payment scenarios and typical constraints in the IoT, and shows that other approaches to digital payments do not fit the IoT use-case well. Section 3 discusses the required functionality to participate as a constrained device in such a system, elaborates on design choices for data transmission formats, and briefly evaluates the proposed design using a proof-of-concept implementation of GNU Taler on RIOT. Section 4 concludes the poster by presenting challenges left for future work.

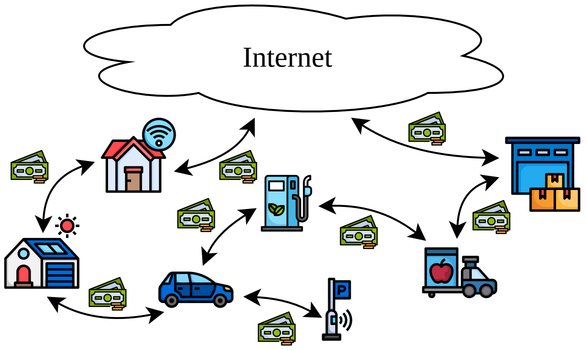

Figure 1: A distributed IoT economy needs autonomous and privacy-respecting machine-to-machine payments.

2. Background and Problem Statement

This section discusses some common IoT scenarios depicted in Figure 1, how these scenarios benefit from IoT-integrated payments, and the requirements and challenges imposed by limited hardware resources. We also analyze currently available or proposed digital payment systems.

IoT Scenarios. The vision of the Internet of Things (IoT) revolves around the seamless cooperation of interconnected devices, operating autonomously without direct human intervention. These devices exchange data or, more generally, services, often involving sensitive information regarding privacy. For instance, smart household appliances such as refrigerators can autonomously order supplies. Industry scenarios may involve the cooperation of many entities, for example, when goods are tracked from manufacturing to warehouse until hand-over to the end-customer, including automatic ordering of new supplies. Similarly, vehicles can autonomously handle payments for parking, tolls, and fuel, benefiting both autonomous and conventional car users. Furthermore, smart grid energy trading relies on IoT devices to coordinate local electricity sharing among buildings equipped with renewable energy sources, ensuring efficient energy management within communities. Even scenarios involving human intervention, such as pay-as-you-go public transportation, can benefit from compact and cheap IoT-based wallets which improve the user experience.

TABLE 1: Comparison of digital payment approaches.

	Approach		
Features	Traditional	Crypto-currencies	Our: E-Cash
Autonomy	✘	✓	✓
Privacy	✘	pseudonymity	payer
Resources	•	••••	••

IoT Payment Requirements. The provision of services (*e.g.,* data sharing) among diverse stakeholders based on an economic incentive model requires economic remuneration. While subscription models suit static scenarios, they fall short in dynamic environments in which IoT devices interact only sporadically. Autonomous machine-to-machine payment might offer a solution by enabling billing and transactions without human intervention. However, concerns about privacy arise due to possible payment observations by third parties, either directly [2] or through metadata analysis [3]. Ensuring payment privacy becomes crucial in scenarios lacking mutual trust among devices. Having an openly standardized privacy-preserving payment system at hand would also counteract monopolies and discriminatory treatment against devices of a certain owner or manufacturer, and allows for true competition and interoperability across devices.

IoT Device Constraints. In scenarios requiring large-scale deployments at minimal cost, devices are typically selected to precisely match the use-case, resulting in a significant number of highly constrained devices. These devices face severe limitations in available memory, including RAM, ROM, and persistent mass storage, which impacts system design in terms of storage requirements and processing overhead. Several IoT scenarios also involve off-the-grid deployments and battery-powered devices, which require the use of low-power wireless networking protocols with small packet sizes and low data rates. A universal IoT payment system must account for these constraints, minimizing storage, processing, and transmission requirements to ensure compatibility with low-end devices.

Payment Options for the IoT. Traditional payment systems, such as credit card payments, bank transfers, and third-party payment providers, are widely utilized by the public for in-store and online transactions. These systems rely on centralized databases storing the account balances, allowing transactions to be initiated through simple means like NFC interactions. However, authentication mechanisms typically require human confirmation, hindering autonomous payments. [4] Moreover, access to the central databases compromises transaction privacy, violating payment privacy requirements. On the other hand, cryptocurrencies offer digital payment alternatives with their decentralized design, seemingly suitable for IoT scenarios. Yet, their reliance on resource-intensive consensus mechanisms and transaction confirmation delays pose challenges. Verifying transactions independently is unfeasible for constrained IoT devices using such approaches. [5] Furthermore, while cryptocurrencies offer pseudonymity, transaction traceability and potential account identity associations remain concerns.

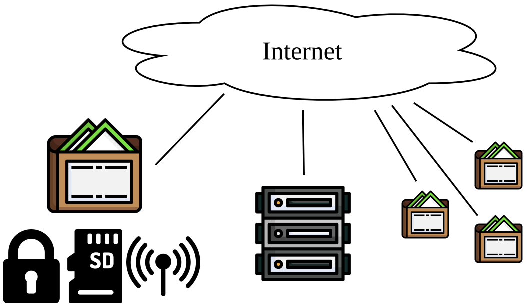

Figure 2: Architecture of an e-cash payment system: Participating devices (▣) need to perform cryptographic operations, store tokens, and communicate with a central service provider (▤) over the Internet.

Our approach: E-Cash. A third approach to digital payments, based on the e-cash scheme pioneered by Chaum [6], offers cash-like anonymity for payers through blind signatures of tokens by a central authority. Each token, signed and backed by a certain value held by the central authority, can be redeemed once by a payee for an authorized payment. Utilizing tokens instead of identity-bound accounts facilitates autonomous operation and hinders transaction linkability. However, the self-custody aspect of e-cash schemes entails token storage requirements for the users, a crucial consideration for deployment on resource-constrained IoT devices. The original design proposed by Chaum could not give unlinkable change and thus had linear complexity for variable amounts. Dold [7] solved this critical issue, allowing for logarithmic complexity in GNU Taler.

Table 1 summarizes our comparison of traditional payment systems, cryptocurrencies, and e-cash concerning the essential requirements identified for the IoT. To the best of our knowledge, this poster represents the first exploration of integrating an e-cash-based payment system with the constrained IoT.

3. Design and Implementation

Design Aspects. Figure 2 shows a typical e-cash system consisting of two basic components: a central *provider*, which issues blind signatures on cryptographic tokens, and *users*, which hold these tokens in self-custody. IoT devices operate as users. Therefore, they need to support three basic functions: (*i*) cryptographic operations such as blinding and signature verification, (*ii*) the storage of signed tokens and metadata, and (*iii*) the communication with the provider via the Internet. We propose to build the payment system integration on top of an IoT operating system with a universal API, abstracting hardware details such as hardware-based cryptographic acceleration, storage, and physical-layer protocols. This approach enables a hardware-agnostic implementation for diverse IoT devices, promoting reusability across deployments.

Currently, our focus has been on efficient data formats for the transmission protocol. In the IoT, reducing packet sizes is crucial because (*i*) low data rates and small Maximum Transfer Units (MTUs) are prevalent in the IoT,

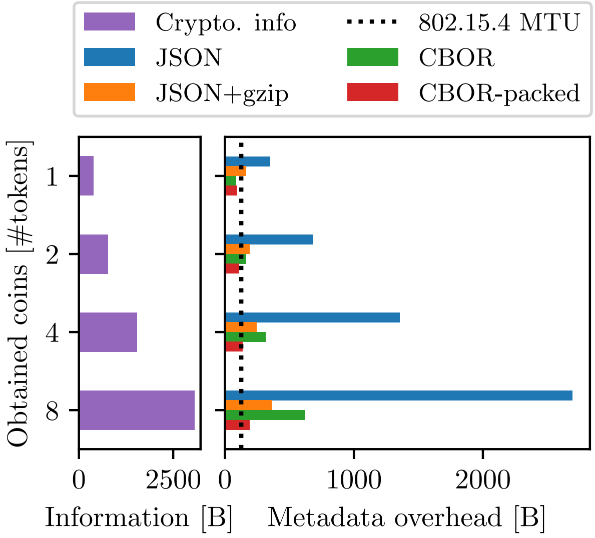

Figure 3: GNU Taler withdraw request payload sizes composed of cryptographic information (left) and metadata overhead depending on the encoding (right): Packed CBOR achieves the smallest format overhead, still fitting one 802.15.4 MTU of 127B for a withdrawal of up to 4 tokens, while avoiding the additional, computationally-intensive compression step of JSON+gzip.

and because (*ii*) fragmentation may lead to additional delay [8]. While the encoding of cryptographic data such as tokens and signatures cannot be reduced below their information content, the accompanying metadata may contain redundant information. Human-readable formats such as JSON or XML may be effective for the broader Internet, where they can be compressed efficiently with methods such as gzip or brotli. Compression, however, introduces complexity on end devices, leading to larger code sizes, higher energy consumption, and additional computation time—all of these characteristics conflict with low-end-device constraints. Introducing a custom binary format containing only the raw cryptographic data in a predefined order is not an option either since it challenges debugging, protocol updates, and forward compatibility. To balance resources and flexibility, we advocate for using CBOR [9], a concise binary data format standardized by the IETF. CBOR efficiently accommodates metadata alongside the data and provides streaming capabilities. Packed CBOR [10], an extension of CBOR, further enhances this efficiency by minimizing metadata redundancy through optimized encoding of repeated information.

Implementation and Evaluation. To evaluate our proposal, we have picked GNU Taler, a free and open-source digital payment system implementing a logarithmic e-cash scheme [7], and RIOT, an free and open-source operating system for the constrained IoT, which provides support for over 270 IoT platforms [11]. Compared to other operating systems, RIOT offers a standardized API for cryptographic operations that can flexibly make use of hardware acceleration and secure key storage where available [12]. The Taler APIs[1] are specified as HTTP-based RESTful protocols using JSON as a data format, with cryptographic data encoded in base32. Figure 3 compares the payload lengths for Taler withdrawal requests encoded in various formats, including JSON, compressed JSON, CBOR, and packed CBOR. Regardless of the number of digital coins acquired, packed CBOR encoding consistently outperforms other schemes, exhibiting approximately half the relative overhead compared to compressed JSON encoding.

4. Conclusion and Outlook

In this abstract, we argued that the IoT will benefit from autonomous and privacy-friendly payments as a common service. Our approach, unlike prior work, suggests a centralized system architecture inspired by the e-cash model. We introduced design choices of our proof-of-concept, which suggest that digital payment is doable even if memory and CPU are constrained. We proposed an efficient standard-compliant data encoding for the communication between user and provider. Future work should focus on token storage efficiency and user-friendly provisioning of IoT devices with digital coins.

References

[1] L. S. Vailshery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030," Jul. 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] J. Lauer, "Plastic surveillance: Payment cards and the history of transactional data, 1888 to present," *Big Data & Society*, vol. 7, no. 1, pp. 1–14, Jan. 2020.

[3] Y.-A. De Montjoye, L. Radaelli, V. K. Singh, and A. S. Pentland, "Unique in the shopping mall: On the reidentifiability of credit card metadata," *Science*, vol. 347, no. 6221, pp. 536–539, Jan. 2015.

[4] M. N. M. Bhutta, S. Bhattia, M. A. Alojail, K. Nisar, Y. Cao, S. A. Chaudhry, and Z. Sun, "Towards Secure IoT-Based Payments by Extension of Payment Card Industry Data Security Standard (PCI DSS)," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–10, Jan. 2022.

[5] S. Mercan, A. Kurt, K. Akkaya, and E. Erdin, "Cryptocurrency Solutions to Enable Micropayments in Consumer IoT," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 97–103, Mar. 2022.

[6] D. Chaum, "Blind Signatures for Untraceable Payments," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Boston, MA: Springer US, 1983, pp. 199–203.

[7] Florian Dold, "The GNU Taler System: Practical and Provably Secure Electronic Payments," Ph.D. dissertation, Université de Rennes, Rennes, France, Feb. 2019. [Online]. Available: https://taler.net/papers/thesis-dold-phd-2019.pdf

[8] M. S. Lenders, T. C. Schmidt, and M. Wählisch, "Fragment Forwarding in Lossy Networks," *IEEE Access*, vol. 9, pp. 143 969–143 987, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3121557

[9] C. Bormann and P. E. Hoffman, "Concise Binary Object Representation (CBOR)," Internet Engineering Task Force, Request for Comments RFC 8949, Dec. 2020.

[10] C. Bormann and M. Gütschow, "Packed CBOR," Internet Engineering Task Force, Internet Draft draft-ietf-cbor-packed-12, Mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-cbor-packed

[11] E. Baccelli, C. Gundogan, O. Hahm, P. Kietzmann, M. S. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch, "RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, Dec. 2018.

[12] L. Boeckmann, P. Kietzmann, L. Lanzieri, T. C. Schmidt, and M. Wählisch, "Usable Security for an IoT OS: Integrating the Zoo of Embedded Crypto Components Below a Common API," *Proc. of 19th International Conference on Embedded Wireless Systems and Networks (EWSN)*, pp. 84–95, 2022.

[1] https://docs.taler.net/core

Poster: FedCM for Research and Education

Erwin Kupris
Munich University of Applied Sciences
Munich, Germany
erwin.kupris@hm.edu

Tobias Hilbig
Munich University of Applied Sciences
Munich, Germany
tobias.hilbig@hm.edu

Thomas Schreck
Munich University of Applied Sciences
Munich, Germany
thomas.schreck@hm.edu

***Abstract*—The misuse of web technologies for tracking users on the Internet poses a threat to user privacy. Such technologies include third-party cookies and bounce tracking servers. Browser vendors and other stakeholders agreed to phase out some of these technologies in the near future. This impacts not only trackers and advertisers, but also legitimate usages such as authentication flows in identity federations. The industry aims to solve these issues via an emerging API called "Federated Credential Management" (FedCM), transforming the login process into a browser-mediated flow. Our research focuses on how to improve the user experience of FedCM within multilateral federations, which are frequently used in the Research and Education (R&E) sector. Specifically, we suggest ways to filter the large number of Identity Providers (IdPs) commonly found in the R&E context and display them, while automating the IdP discovery process. We provided our suggestions to the working group, also considering user privacy aspects. Incorporating these changes into the FedCM API accordingly could pave the way for a privacy-preserving and user-friendly sign-in experience in R&E federations.**

***Index Terms*—FedCM, federation, security, privacy, academia**

I. Introduction

The ongoing exploitation of certain web technologies for tracking poses a threat to user privacy and data protection. To mitigate these infringements, browser vendors have agreed to phase out some of these technologies. The most prominent example is third-party cookies, which are expected to be deprecated later this year. Bounce tracking, i.e., a specific chain of redirects invisible to the user, serves similar purposes. However, legitimate usages for these technologies do exist, which will be affected by this discontinuation.

Federated authentication mechanisms fall into this category. SAML2 and OAuth are the most commonly used protocols for this purpose. Both employ redirect-based login flows that are nearly indistinguishable from malicious usage. Moreover, federated applications often use services such as SeamlessAccess for discovering a user's Identity Provider (IdP) [1]. To offer a comfortable user experience, third-party cookies are required.

To prevent the breakage of these technologies, the Federated Credential Management API (FedCM) was proposed. It enables a secure, privacy-preserving, and dynamic authentication process mediated by the browser. FedCM is being developed by the W3C and has the status of a draft community group report. A working group with the goal of standardizing this API was recently established. The latest draft of the FedCM API is already implemented in the Chrome browser and experimental features can be tested in the Canary version.

In FedCM, the browser acts as a mediator between Relying Parties (RPs) and IdPs [2]. When a user visits an RP, it can call the FedCM API by providing one or more IdP URLs. The browser then issues requests to the IdP to obtain information about user accounts that have an active session with that IdP. Afterwards, the browser asks the user to select an account and consent to the federated authentication in a mediated dialog. Upon selection, the browser issues a final request to the IdP that includes the session cookie of the chosen account. The IdP returns an opaque identity assertion to the browser, which is relayed to the RP, thereby concluding the FedCM flow.

The current version of the FedCM API covers use-cases of social, bilateral federations, e.g., "Login with Google" or "Login with Facebook". However, multilateral federations, common in Research and Education (R&E), typically have different requirements. Although FedCM does not yet meet these requirements, it has the potential to not only fulfill them but also address other long-standing issues.

In this work, we propose ideas for extending FedCM to function effectively in the R&E sector. They can also be applied to other federated architectures such as Open Banking.

II. Problem Statement

R&E federations differ from bilateral federations by having the following specific characteristics: Instead of a single or a few public IdPs, users from R&E institutions can usually choose from thousands of IdPs when authenticating to an RP. Due to this extensive number of IdPs, "Where Are You From" (WAYF) services are used to determine a user's home IdP. Moreover, session lifetimes are commonly shorter, e.g., about an hour long. R&E IdPs usually do not offer a dedicated login page. Instead, when accessing a compatible service, users are redirected to the IdP for authentication. Finally, it is possible for RPs to not be explicitly registered at each IdP. These differences prevent the latest version of the FedCM API from being used in R&E federations.

Our contributions in this poster focus on two aspects of FedCM: (1) FedCM should only present compatible IdPs to the user, i.e., IdPs accepted by the RP and affiliated to the user. (2) FedCM should provide an enhanced user experience when the API is called using multiple, multilaterally federated IdPs. This would enable an automated, user-friendly WAYF process for R&E federations. In addition, we discuss privacy aspects that become relevant when extending FedCM to better support multilateral federations.

III. Filtering IdPs

In the early stages of FedCM, an RP could initiate the API using only a single IdP per call. However, RPs often provide federated authentication with multiple providers. This requires RPs to display individual buttons for each supported IdP, initiating FedCM for the chosen IdP. This is known as the so-called *NASCAR* problem, because users are presented with an overwhelming number of logos and buttons.

To avoid this issue in FedCM, two experimental features have already been added to the API: (1) A list of IdPs might be provided in the API call instead of a single one [3]. (2) IdPs can register themselves in the user's browser [4]. In the so-called *any* mode, RPs can then call FedCM for all registered IdPs without explicit knowledge about them. While still being actively worked on, both of these features enable RPs to initiate FedCM using a larger number of potential IdPs.

However, there are many scenarios in which only IdPs suitable for federated authentication with a specific RP should be considered in the FedCM procedure. For example, an RP might only be federated with IdPs that are part of the global R&E inter-federation eduGAIN. In this case, other IdPs should not be considered during the FedCM procedure to avoid unnecessary requests by the browser. This section provides suggestions for realizing this functionality.

A. IdP-list approach

RPs in R&E federations can usually fetch lists of compatible IdPs from a federation operator or a separate metadata discovery service. With this approach, an RP can already initiate FedCM using only IdPs it is federated with, omitting any unsuitable ones. While this solution is straightforward, it presents another challenge: A list of all IdPs with which an RP is federated might be extensive, e.g., there are more than 5,000 possible IdPs in eduGAIN. It is not practical for the browser to issue requests to all of these IdPs. Therefore, the following approaches might be considered to filter such extensive lists provided by the RP.

FedCM introduced the Login Status API to prevent unnecessary requests to IdPs. This API enables IdPs to set a status, either *logged-in* or *logged-out*, for their domain within the browser. Before initiating any requests to an IdP, the browser checks that the status for that IdP is set to *logged-in*. If the RP provides multiple IdPs, this verification process occurs concurrently for each. Consequently, only IdPs that have set this status, i.e., IdPs with which the user has previously interacted, are considered by the browser.

This solution works well for social federations because session lifetimes are typically very long. In other scenarios, however, sessions are usually shorter, especially in the academic or banking sectors. Such IdPs can utilize this API by consistently setting the status to *logged-in*, preventing them from being filtered out by the browser if the user is not logged into them. While this solution might be perceived as misusing the API, it would work with the current FedCM implementation in Chrome without necessitating any changes.

Instead of utilizing the login status API to infer with which IdPs the user has already interacted, FedCM's IdP registration can be used to filter IdP lists sent by an RP. In the proposed *any* mode, every IdP that has been previously registered in the browser is considered for the regular FedCM flow. Because the RP does not call the API with IdPs that it is compatible with, this mode needs to be adjusted to not consider unsuitable IdPs that might have registered. Such an adjustment can combine the IdP registration with lists of IdPs sent by RPs during the API call, resulting in a *some* or *certain* mode. If an IdP has been previously registered in the browser and is included in the list of compatible IdPs provided by the RP, it can reasonably be assumed that it is suitable for federated authentication. Therefore, such an IdP should be considered by FedCM, regardless of its login status.

B. affiliationHint approach

Instead of the RP sending a whole list of IdPs, we suggest it signals its federation affiliations. This could be realized via an additional attribute, e.g., *affiliationHints*, that the RP provides to FedCM. Similarly, the IdP would mark its federation affiliations during IdP registration. When the API is called, the browser compares IdPs registered with identical *affiliationHints* and exclusively considers exact matches. Since R&E federations are often structured hierarchically, this attribute can contain multiple entities, as shown in the following example.

The Munich University of Applied Sciences (HM) is a member of the German federation DFN-AAI and the eduGAIN inter-federation. HM's IdP registered itself in the browser with *affiliationHints = [”hm.edu”, ”dfn.de”, ”edugain.org”].* The user accesses an RP operated by the Sapienza University, which is a member of the Italian GARR and eduGAIN. The RP sends *affiliationHints = [”uniroma1.it”, ”garr.it”, ”edugain.org”]* within the API call. The browser then determines that RP and IdP share an affiliation, i.e., eduGAIN. Afterwards, the regular FedCM flow continues with HM's IdP.

If an entity is part of more federations, this list can be simply extended. In scenarios where such a clear hierarchy does not exist, other hints might be used, e.g., *”EU-bank”* or *”US-bank”*. The exact structure of these hints must be clearly defined to ensure alignment between RPs and IdPs. Furthermore, both RPs and IdPs can restrict their *affiliationsHints* as they desire. For example, by excluding eduGAIN from its hints, an RP can ensure that only users within its own national federation are presented with an option to access it via FedCM.

C. OpenID Federation Approach

Instead of *affiliationHints*, the RP can call FedCM by including OpenID Federation trust chains [5]. Each trust chain represents a signed path from the RP to one of its trust anchors. For filtering IdPs, the *entityIds* within a trust chain can be parsed and used similarly to the *affiliationHint* approach. The RP's trust chains can subsequently be used to verify the trust relationship between the RP and IdP. This approach is similar to our previous study, which proposes a way to automate the IdP discovery process in multilateral federations [6].

IV. WAYF operation mode

As previously discussed, users within R&E federations can authenticate at federated RPs via thousands of potential IdPs. Consequently, RPs in such federations commonly integrate WAYF services to determine the IdP with which the user wants to authenticate. This process is cumbersome from a user experience perspective, because it frequently involves selecting one's home organization from an extensive list. Additionally, the user experience of WAYF services is further impaired by the deprecation of third-party cookies. The emerging FedCM API has the opportunity to automate this long-standing issue within R&E federations.

When the FedCM API is called by the RP, it currently offers the user a selection of logged-in accounts. We propose FedCM to incorporate an "organization chooser" in addition to the current account chooser dialog. This dialog leverages some of FedCM's experimental features, i.e., IdP registration and "button mode". Instead of showing only logged-in accounts, the selection dialog should include organizations that have registered themselves in the browser and are not filtered out by the methods proposed in Section III. If the user selects an organization, a subsequent login should be facilitated at the IdP by opening a pop-up window at the login URL the IdP previously registered with. However, R&E IdPs might not allow users to authenticate at the IdP directly. Instead, the federated login procedure can only be initiated via a redirect from an RP, including the necessary parameters such as the RP's entity identifier. Therefore, R&E IdPs will need to develop alternative solutions, e.g., integrating a separate RP at the login URL.

If FedCM were to be adopted in R&E federations in the future, changes to all involved IdPs and RPs would be necessary. During the transition period, it would be beneficial for FedCM to support basic WAYF functionality. Normally, FedCM requests an opaque token from the IdP for the selected account and returns it to the RP. Instead of returning a token, we suggest that FedCM should offer an option to return the IdP selected by the user. This can be realized via an additional parameter called *wayf* that is set by the RP in the API call. Upon selection, the browser skips the retrieval of the token from the assertion endpoint. After the selected IdP is returned, the existing, possibly redirect-based federated login flow is executed. Realizing this functionality would require minimal changes to the affected RPs, IdPs, and the FedCM API.

V. Privacy Considerations

After a list of logged-in accounts has been queried, the FedCM flow continues by fetching client metadata about the RP from the IdP. Apart from receiving the RP's metadata, this request also ensures that a trust relationship between IdP and RP exists. In the public IdP use case, RPs are always registered at the IdP, making such a request possible. However, in multilateral federations, this is often not the case. Instead, metadata is either centrally managed, for example in SAML2, or resolved dynamically in OpenID Federation.

In SAML2 based federations, the IdP typically maintains and regularly updates extensive XML files containing the metadata of all RPs within a federation. Therefore, the IdP can simply locate the RP's metadata in these files and return it. For federations based on the OpenID Federation protocol, this process presents a potential privacy challenge. It is vital for FedCM to never disclose the user's affiliation, i.e., their IdPs, to an RP before they give explicit consent. The regular metadata resolution in OpenID Federation would violate this rule if the IdP started this process with a request to the RP [5]. A malicious RP can correlate such a request with the user's running browser session and infer the user's affiliation. In our previous work, we presented an alternative approach to verifying the trust relationship between IdP and RP [6]. This method requires the RP to initiate the FedCM API by including its OpenID Federation trust chains, as stated in Section III-C. In addition, this solution does not disclose the RP a user visits to the IdP before the user consents.

VI. Conclusion and Future Work

The Federated Credential Management API (FedCM) is an emerging standard that aims to improve privacy, security, and the overall user experience of authenticating to federated webservices. In the R&E sector, specific requirements exist that FedCM, as of today, does not fully cover. Our poster presents ideas on how this API can be extended to better support the R&E sector and to offer a user friendly sign-in experience. We have already proposed these suggestions to the working group [7].

In future work, we plan to build a proof of concept of FedCM at our university's IdP. This includes implementing a FedCM plugin for the Shibboleth IdP software and integrating the necessary API endpoints. Furthermore, we plan to analyze its security through a threat model analysis. Finally, we envision a usability study representative of higher education institutions, including students, staff, and faculty members.

Acknowledgments

We are grateful for having had the opportunity to discuss our initial idea for this work at TDI / OSW 2024, particularly with Giuseppe De Marco, Tim Cappalli, and Samuel Goto.

References

[1] Coalition for Seamless Access, "SeamlessAccess," 2024. [Online]. Available: https://seamlessaccess.org/

[2] N. P. Moreno, "Federated Credential Management API," W3C, Draft Community Group Report, Mar 2024. [Online]. Available: https://fedidcg.github.io/FedCM/

[3] W3C FedID CG, "Allow multiple idps to be used," GitHub Issue, 2022. [Online]. Available: https://github.com/fedidcg/FedCM/issues/319

[4] T. Looker et al., "Allow IDP registration," GitHub Issue, 2023. [Online]. Available: https://github.com/fedidcg/FedCM/issues/240

[5] R. Hedberg, M. B. Jones, A. A. Solberg, J. Bradley, G. De Marco, and V. Dzhuvinov, "OpenID Federation 1.0 - draft 34," The OpenID Foundation, 2024.

[6] E. Kupris, T. Hilbig, D. P. Sugar, and T. Schreck, "A-WAYF: Automated Where Are You From in Multilateral Federations," in *2nd International Workshop on Trends in Digital Identity (TDI 2024)*, 2024.

[7] E. Kupris and T. Hilbig, "FedCM for Research and Education," GitHub issue, 2024. [Online]. Available: https://github.com/fedidcg/FedCM/issues/563

Poster: Empirical Cybersecurity Investment Decision-Making: Bridging the Gap Between Intuition and Metric-Driven Strategy

Nadia Lorraine Niyonsaba
University of Twente
Enschede, Netherlands
n.n.l.niyonsaba@utwente.nl

Abhishta Abhishta
University of Twente
Enschede, Netherlands
s.abhishta@utwente.nl

Jeroen van der Ham-de Vos
University of Twente
Enschede, Netherlands
j.vanderham@utwente.nl

Laura Spierdijk
University of Twente
Enschede, Netherlands
l.spierdijk@utwente.nl

***Abstract*—Organizations are under constant threat of cyber attacks. Hence, they need to have information security measures put in place. Security managers lack methods to quantify how much effort should be invested in protecting their IT assets and training people. We propose developing a metric-based method to guide cybersecurity investments, aimed at improving an organization's security posture.**

I. Introduction

Cybersecurity managers lack clear metrics to guide security decisions especially when it comes to investing in security[1]. Optimal security investments are paramount as organizations increasingly rely on Information Technology (IT) [2]. However, organizations rarely have insights into the potential added value offered by security investments. In 2016 RedSeal Inc. polled 200 CEOs; 87% of CEOs who were surveyed expressed the need for better metrics to evaluate cybersecurity investments, and 72% said that the lack of useful metrics for cybersecurity investment assessment was a "major challenge" [3], [4]. [5] and [6] proposed security investment models, but the models have limitations, including being based on security threats and vulnerabilities that are hard to quantify [6], [7], and lacking the organization security context[1], [8]. This research proposes a new decision-making process that is metric-centered and tailored to each organization's security posture [9], [10].

Typically, organizations use different metrics to quantify and justify their decisions with several business operations but do not follow the same rational decision-making approach to allocate resources for cybersecurity in their organizations[1], [11]. The lack of proper resource allocation for cybersecurity makes information security passive and less important for most organizations, even though one single attack can be fatal and lead to business closure. For example, in 2021, a major gas pipeline (Colonial Pipeline) endured a ransomware attack that led to paying a ransom worth $4.4 million, following the attack. The attack also affected some of the pipeline's digital programs, shutting it down for several days[12]. Organizations use methods like discounted cash flow, Monte Carlo simulation, decision-tree analysis, and others for uncertain and complex decisions[13]. Nevertheless, interviews conducted by [11] reveal that most security executives and other organization managers do not use any investment quantification methods to decide how much their organizations should invest in cybersecurity.

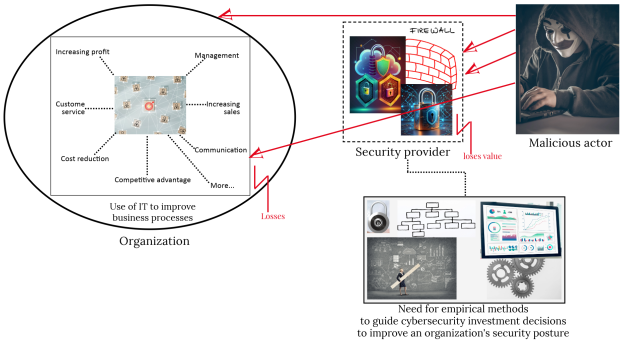

Fig. 1. Situation and complication - Information Technology adds value to business operations leading to success, however, cyberattacks use the same technology as a vector to harm organizations when there are no security measures put in place.

Organizations using information technology must implement cybersecurity measures to safeguard their IT assets and data. The increasing reliance on IT for competitive advantage [14] underscores the necessity of cybersecurity. Cyberattacks lead to consequences such as financial losses, reputational damage, and sometimes business closure; emphasizing the importance of effective cybersecurity measures [2]. For example, an Illinois hospital closed in 2021 after failing to financially recover from a ransomware attack[15]. Frequently, most organizations don't put enough information security measures or are passive to protect their IT assets from attacks, due to lack of awareness and lack of proper decision tools and models to quantify how much they should invest in cybersecurity.

We propose developing a metric-based method to calculate cybersecurity investments, tailored to each organization's security posture.

[5], [6] proposed cybersecurity investment models, but the models are uncommon and have limitations [1], [7], [11]. For example, the Gordon and Loeb information security in-

vestment model [5] suggests that businesses should invest not more than 37% of the expected loss that would result from potential attacks on that business's information. But, quantifying threats is still a challenge[6]. For example, an organization cannot quantify zero-day vulnerabilities [16]. In addition, [6] proposed that organizations combine different investment quantification methods, instead of just one, because one investment method would be impractical. However, the method has limitations [6], [7] and cannot be used as [7] shared after analyzing that proposed approach. Therefore, more research is required to propose usable quantitative cybersecurity decision-making models for investment and other decisions.

Despite the rare use of metric-centered methods for cybersecurity decisions, organization directors must protect their businesses from cyber threats and have different approaches to it. Mental biases and social preferences are some of the frequent factors that influence cybersecurity decisions[17]. However, expert cybersecurity decision-makers depend on security frameworks like NIST-800-53 and ISO 27001 to structure strategic security decision-making processes [18]. Nonetheless, [17] states that even experts can exhibit errors and have difficulties understanding delays and uncertainties in predicting cyber incidents. Besides security experts, [19] shares that most managers who have to make cybersecurity decisions lack the expertise to make information security decisions. Therefore, quantitative methods must be used to support rational decisions in information security, especially in uncertain and unfamiliar incidents.

II. Security Posture based investment

We propose developing a new cybersecurity investment measuring method that is based on every organization's context (security posture) [9], [10]. We also propose a framework that organizations can use to measure their security posture, which will then inform different security metric-based decisions, including investment decisions. Rowe & Galler[1] and Vries[8] have proposed a conceptual approach to describe and consider the components of a cybersecurity investment decision. Following their idea, we will explore an organization's security posture as the base for identifying where an organization stands with information security; what, and how much the organization should do to improve its information security posture.

To arrive at our proposed research goals, we have three main research questions whose answers will be building blocks to achieving our research targets. Our first research question is "Why should organizations care about information security investment?" To properly answer it we have a couple of sub-questions which are: "What are the comprehensive direct and indirect costs associated with cyberattacks on organizations, and how can these costs be efficiently quantified and assessed?", "How has cybersecurity investment been defined from an economic perspective, in published literature?", and "How do organizations currently make decisions related to cybersecurity investment?". The second research question is "How can an organization's security posture be used for decision-making?" To answer it we have a couple of sub-questions which are: "How has security posture been defined in published literature, under the scope of cybersecurity decision-making?", "What variables should be used to measure and improve an organization's security posture?" and "How can we systematically measure and evaluate an organization's security posture based on our understanding of the factors that contribute to it?". Lastly, answering the third question which is: " How can an organization's security posture be used to calculate how much to invest in information security?" will lead us to achieving the final research goal.

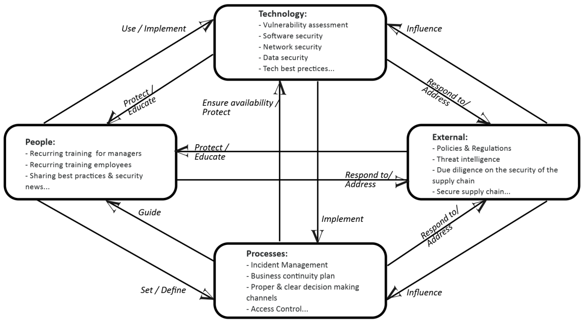

Fig. 2. Assumption: Elements of an organization's security posture. An organization's security posture is made of different factors from the people within an organization, to policy and regulations, the technology they use, and external threats and opportunities that affect its cybersecurity. Our research aims to bring together all those factors together into a framework that will guide organizational security decisions.

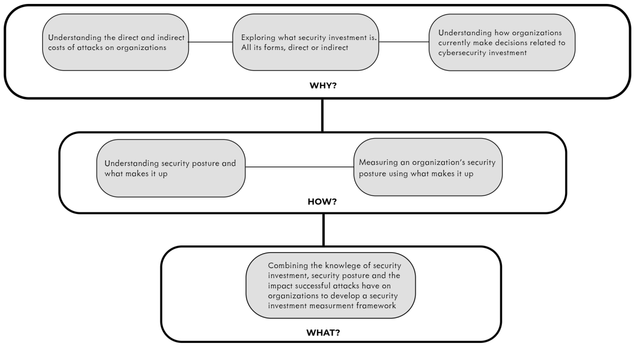
Understanding the direct and indirect costs of attacks on organizations
Exploring what security investment is. All its forms, direct or indirect
Understanding how organizations currently make decisions related to cybersecurity investment
WHY?
Understanding security posture and what makes it up
Measuring an organization's security posture using what makes it up
HOW?
Combining the knowlege of security investment, security posture and the impact successful attacks have on organizations to develop a security investment measurment framework
WHAT?

Fig. 3. Our research approach

III. conclusion

To determine the best value for cybersecurity investment organizations must use a framework that considers the context of each organization. Existing decision-making frameworks and security investment models do not consider existing or lacking effort in building an organization's security posture, the models base their calculations on security threats even though the threats are hard to quantify and the approach leaves out each organization's context. To address the mentioned limitations metric-centered methods for information security decisions and investment, we aim to develop an organization security posture tailored decision-making method that will also be used to calculate cybersecurity investments within organizations. Therefore, we call fellow researchers participating at EuroS&P 2024 and beyond, who are interested in research to develop a usable investment cybersecurity decision-making method to collaborate with us in addressing the stated research questions. Together, we can pave the way for a more resilient and proactive cybersecurity landscape, ensuring the sustained success of organizations in an increasingly digitized world.

References

[1] B. R. Rowe and M. P. Gallaher, "Private sector cyber security investment strategies: An empirical analysis," in *The fifth workshop on the economics of information security (WEIS06)*, 2006.

[2] A. Kazemi, M. Kalhornia Golkar, and S. Lajmiri, "Origins of cyber security," *International Journal of Reliability, Risk and Safety: Theory and Application*, vol. 6, no. 2, pp. 77–83, 2023.

[3] R. A. Rothrock, J. Kaplan, and F. Van Der Oord, "The board's role in managing cybersecurity risks," *MIT Sloan Management Review*, vol. 59, no. 2, pp. 12–15, 2018.

[4] I. RedSeal, *Ceos reveal cyber naivete as incidents rise and losses mount*, en, Dec. 2016. [Online]. Available: https://www.globenewswire.com/en/news-release/2016/12/13/1194613/0/en/CEOs-Reveal-Cyber-Naivete-as-Incidents-Rise-and-Losses-Mount.html.

[5] L. A. Gordon and M. P. Loeb, "The economics of information security investment," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 438–457, 2002.

[6] B. Jerman-Blažič *et al.*, "An economic modelling approach to information security risk management," *International Journal of Information Management*, vol. 28, no. 5, pp. 413–422, 2008.

[7] L. Podešva and M. Koch, "Comparison of the most important models of investments in cyber and information security," *Trends Economics and Management*, vol. 16, no. 39, pp. 25–34, 2022.

[8] J. de Vries, "What drives cybersecurity investment?: Organizational factors and perspectives from decision-makers," 2017.

[9] C. C. NIST Editor, *Security posture - glossary*. [Online]. Available: https://csrc.nist.gov/glossary/term/security_posture.

[10] D. U. Tran and A. Jøsang, "Information security posture to organize and communicate the information security governance program," in *European Conference on Management Leadership and Governance*, vol. 18, 2022, pp. 515–523.

[11] T. Moore, S. Dynes, and F. R. Chang, "Identifying how firms manage cybersecurity investment," in *Workshop on the Economics of Information Security (WEIS)*, 2016, pp. 1–27.

[12] S. M. Kerner, "Colonial pipeline hack explained: Everything you need to know," *TechTarget, April*, vol. 26, 2022.

[13] L. Trigeorgis, *Real options: Managerial flexibility and strategy in resource allocation*. MIT press, 1996.

[14] J. G. Mooney, V. Gurbaxani, and K. L. Kraemer, "A process oriented framework for assessing the business value of information technology," *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, vol. 27, no. 2, pp. 68–81, 1996.

[15] K. Collier, *An illinois hospital is the first health care facility to link its closing to a ransomware attack*, en, Jun. 2023. [Online]. Available: https://www.nbcnews.com/tech/security/illinois-hospital-links-closure-ransomware-attack-rcna85983.

[16] Kaspersky, *What is a zero-day attack? - definition and explanation*, Apr. 2023. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/zero-day-exploit.

[17] M. S. Jalali, M. Siegel, and S. Madnick, "Decision-making and biases in cybersecurity capability development: Evidence from a simulation game experiment," *The Journal of Strategic Information Systems*, vol. 28, no. 1, pp. 66–82, 2019.

[18] B. Shreeve, J. Hallett, M. Edwards, P. Anthonysamy, S. Frey, and A. Rashid, "“so if mr blue head here clicks the link..." risk thinking in cyber security decision making," *ACM Transactions on Privacy and Security (TOPS)*, vol. 24, no. 1, pp. 1–29, 2020.

[19] B. Shreeve, C. Gralha, A. Rashid, J. Araujo, and M. Goulão, "Making sense of the unknown: How managers make cyber security decisions," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 4, pp. 1–33, 2023.

Poster: Byzantine Fault Tolerant State Machine Replication for Non-deterministic Applications

1st Pedro Camponês
Department of Informatics
NOVA University Lisbon
Lisbon, Portugal
p.campones@campus.fct.unl.pt

2nd Diogo Tavares
Department of Informatics
NOVA University Lisbon
Lisbon, Portugal
dc.tavares@campus.fct.unl.pt

3rd João Magalhães
Department of Informatics
NOVA University Lisbon
Lisbon, Portugal
jmag@fct.unl.pt

4th Henrique Domingos
Department of Informatics
NOVA University Lisbon
Lisbon, Portugal
hj@fct.unl.pt

***Abstract*—Byzantine Fault Tolerant State Machine Replication (BFT-SMR) is a widely used approach to provide availability and fault tolerance to applications by enabling the system to continue its correct execution even when some nodes present arbitrary faults. This is done by replicating for all nodes the execution and response to client requests. When the client receives enough identical responses, that response is delivered. In BFT-SMR all correct processes must produce equal responses to the client requests. This requisite hinders the use of Large Language Models (LLMs) given that these models inherently produce non-deterministic outputs. We introduce Model Answer Replication (MARS), an algorithm that allows the use of SMR in non-deterministic applications, particularly when employing LLMs. MARS' key innovation is comparing responses to client requests by measuring their semantic similarity rather than bit by bit equality and allowing client side code to remain unaltered from what is found in SMR algorithms for deterministic applications.**

***Index Terms*—State Machine Replication, Byzantine Fault Tolerance, Large Language Models, Non-deterministic Algorithms**

I. Introduction

Instruction-tuned Large Language Models (LLMs) are deep-learning models designed to generate natural language textual responses to comply with user requests. Their success has enabled their integration into diverse applications, showcasing their utility across multiple domains [1], including critical applications such as medical diagnostics [2].

In critical applications, the safety and availability of the system is crucial. One common method to achieve this is through the Byzantine Fault Tolerant [3] State Machine Replication [4] (BFT-SMR) abstraction, where the components of the system are replicated. When a client issues a request, all nodes will execute it and send their response to the client, ensuring the system can execute correctly even when some nodes fail.

With BFT-SMR, when a client makes a request, they receive multiple responses. If enough responses match, the system considers the request successful. This method clashes with LLMs, as their responses vary due to the non-determinism commonly introduced during language decoding. Existing solutions to abstract non-determinism in applications and allow their use in the SMR approach [5], [6] fail to extend to the challenges posed by LLMs.

We introduce Model Answer Replication (MARS), an algorithm designed to make LLMs compatible with SMR. To the best of our knowledge, this work represents the first attempt to address this issue. A key innovation of MARS is to compare server responses based on their semantic/conceptual similarity rather than by exact match. Furthermore, MARS and BFT-SMR protocols for deterministic operations differ in that in the latter the response the client delivers is the output of correct nodes, while in MARS, the response delivered can result from a byzantine node, as long as it is be deemed innocuous by correct nodes. This abstract outlines our ongoing work. We will:

- Using fundamental concepts in BFT-SMR and unsupervised learning as basis (§ II) and under a realistic system model (§ III), we will specify (§ IV) and implement the MARS algorithm.
- Validate and evaluate a critical application using MARS on a representative environment with heterogeneity in the components used and under diverse attacks to the system safety.

II. Background

In this section, we describe the primitives used in MARS, adapting the definitions from [7].

Best Effort Broadcast (BEB) is a simple broadcast primitive which satisfies the following properties:

- **Validity (BEB1)**: If a correct process broadcasts *msg*, every correct process eventually delivers *msg*.
- **No duplication (BEB2)**: No message is delivered more than once.
- **No creation (BEB3)**: If a process delivers *msg* with sender *from*, then *msg* was previously broadcast by *from*.

Total Order Broadcast (TOB) is a broadcast primitive that establishes an order for messages to be delivered, followed by all correct processes in accordance with these properties:

- **Validity (TOB1)**, **No duplication (TOB2)**, and **No creation (TOB3)**: Equal to properties BEB1, BEB2, and BEB3, respectively.

This work was supported by the FCT/MCTES grants PRT/BD/152803/2021 and PRT/BD/154428/2022, as well as by NOVA LINCS UIDB/04516/2020 and UIDP/04516/2020 with the financial support of FCT.IP

- **Agreement (TOB4)**: If *msg* is delivered by a correct node, then it is eventually delivered by all correct nodes.
- **Total Order (TOB5)**: Let *msg1* and *msg2* be any two messages and suppose p and q are any two correct processes that deliver *msg1* and *msg2*. If *p* delivers *msg1* before *msg2*, then *q* delivers *msg1* before *msg2*.

Mapping to Embedding Space is the process of mapping a sentence in natural language into an *n*-dimensional point [8]. To implement this process, a Transformer encoder [9] identifies key aspects of natural language sentences such that similar sentences map to close points in space.

III. System Model

Service Properties: We assume a system with a static number of *n* nodes. All nodes have the same state but may be running heterogeneous implementations of the system's specification [5]. MARS provides the State Machine Replication (SMR) abstraction [4] and the following properties:

- **Linearizability (S1):** The execution of client requests should behave as a centralized component [10];
- **Pragmatic Correctness (S2):** Every response a client delivers is valid and results from a correct node, or is a similar and equally valid response from a faulty node;
- **Termination (L1):** A request from an honest client will eventually be executed and its response delivered;

The first two properties are safety properties, Termination is a liveness property. Properties S1 and L1 follow directly from the requirements to provide the SMR abstraction [4]. Pragmatic Correctness ensures that the response the client delivers is correct, even though MARS does not guarantee the clients deliver responses from correct nodes.

Network Model: We assume the adversary controls the network and is able delay and reorder messages exchanged between all parties, controlling the scheduling of messages.

MARS' execution requires calls to a Total Order Broadcast (TOB) primitive. All steps of the MARS algorithm excluding the TOB calls can be executed in an asynchronous network model, where the adversary can delay messages an indefinite (but finite) amount of time. The network model is therefore limited by the underlying TOB primitive implementation, which can be either a partially synchronous network [11] with deterministic BFT-SMR protocols [5], [12] or an asynchronous network with randomized protocols [13].

Adversary Model: We assume a static adversary that can induce byzantine faults [3] in at most *f* nodes, where $f < \frac{n}{3}$. The adversary is computationally bounded and thus unable to subvert standard cryptographic primitives except with negligible probability. Furthermore, we assume that all parties involved are authenticated and sign their message contents.

IV. MARS

Figure 1 represents an execution trace of MARS responding to a client request assuming there are four nodes and the second is byzantine. The client request will be totally ordered using the Total Order Broadcast (TOB) primitive. Each node runs its own LLM, and each of which will produce a different response to the client request, however, correct nodes will produce semantically similar responses. In fig. 1 correct nodes provide common greetings (*Hello*, *Hi*, and *Howdy*). Faulty nodes produce arbitrary responses that may or may not be similar to correct nodes', *e.g* the response *Bye*.

After generating the responses to the client request, the nodes choose which response is sent to the client, while excluding incorrect responses. This is achieved following two steps: Nodes evaluate each other's responses and group similar responses.

Entailment Evaluation: First, nodes broadcast their responses through the Best Effort Broadcast (BEB) primitive so they are delivered by all correct nodes. Each node evaluates a peer response by performing Natural Language Inference (NLI) to measure the *entailment* between the client request and the response being evaluated [14]. A high score indicates strong entailment while a low score indicates there is no relation between the request and the response.

Response Set Consensus: Correct nodes then use BEB to broadcast the scores of each response evaluated. Following property BEB1, every correct node *i* will eventually aggregate a set S_i of $n - f$ responses and $2f + 1$ scores per response. Properties BEB2 and BEB3 assure that the majority of responses and scores result from correct nodes. Only one set S_i will be used to decide the response sent to the client by correct nodes. Every set S_i will have a majority of correct responses and scores and so any can be chosen. Nodes will propose their sets S_i and we use TOB to achieve consensus on the set *S* to be used [15]. Properties TOB1, TOB4 and TOB5 ensure all correct nodes deliver and agree on *S*.

Answer Selection: Each correct node runs the deterministic algorithm *Answer Selection* (AS) on input *S*, which selects a single response from *S* to be sent to the client. In AS, every response is mapped to a point in the embedding space [9], such that semantically similar responses produce points that are closer than dissimilar responses. Byzantine responses that are semantically different from correct responses map to points far from correct responses. However, experimentally we obverse that there are scenarios where this behaviour is not present. To account for this phenomena, we use the NLI scores gathered in *S* to spread apart points with lower entailment.

For each response, its final score will be the median of its corresponding scores in *S*. Because there are $2f+1$ scores per response, where at least $f + 1$ result from correct nodes, the median score will either have been proposed by a correct node or has a value between scores proposed by correct nodes. A point corresponding to a low score response will be displaced further away from the origin, while a point corresponding to a high score response will not be displaced. Because there are at most f byzantine responses in S and at least $f + 1$ correct responses, we perform agglomerative clustering, incrementally adding new points until a cluster of size at least $f + 1$ is achieved. The chosen response will correspond to the point that minimizes the square root of the distance to the remaining

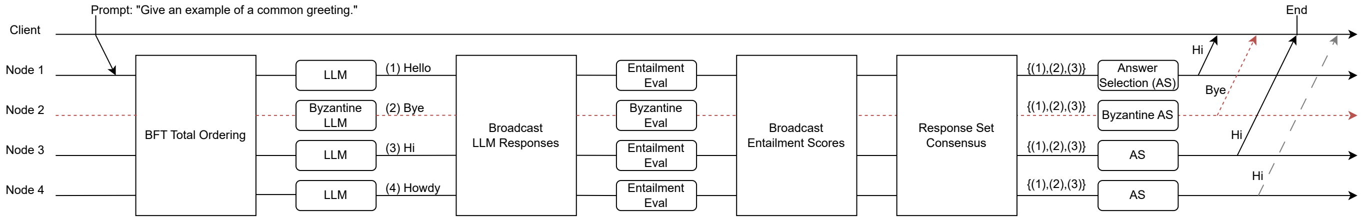

Fig. 1. Execution trace of a idempotent client request in MARS when Node 2 is Byzantine. The client issues a request, which is ordered so all correct nodes process it in the same relative order. Passing the request by a correct LLM results in a variation of a greeting (”Hello”, ”Hi”, and ”Howdy”). Nodes broadcast their responses and each will evaluate the responses of its peers. The evaluation scores will then be broadcast and nodes will collect $2f+1$ scores for $n-f$ responses. Nodes will then determine the set of candidate responses, along with their scores. Nodes then select a response and send it to the client. When the client receives $f+1$ equal responses (”Hi”), the content of these responses is delivered.

points in the cluster, resulting in the selection of the innermost point while discounting the influence of points in the edge of the cluster. All correct nodes will compute the same response from the AS algorithm and send it to the client. The client delivers a response that it has received from at least $f+1$ nodes.

Properties: The *Linearizability* property is achieved by ordering the client request through the use of the TOB primitive a the start of MARS execution. The nodes in the MARS algorithm execute calls to LLMs, the *Entailment Evaluation*, and *Answer Selection* procedures, all of which are local and always terminate. Besides these procedures, the nodes make calls to the BEB and TOB primitives. Properties BEB1, TOB1, and TOB4 ensure that these protocols always terminate. It follows that MARS achieves its *Termination* property.

Finally, it is not obvious MARS achieves the *Pragmatic Correctness* property. In our model we assume that responses by byzantine nodes map to points that are either (1) far apart from points resulting from correct nodes, if the byzantine responses are malicious; or (2) are close to points proposed by correct nodes but have innocuous content. A more formal definition of what constitutes innocuous content in our model will be present in the full version of this work. Experimentally, we find this assumption to hold true in most scenarios; however, a more thorough evaluation and mechanisms to better approach our model to reality are still in progress.

References

[1] M. U. Hadi, q. a. tashi, R. Qureshi, A. Shah, a. muneer, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, and S. Mirjalili, “Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects,” 2023.

[2] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal, M. Schaekermann, A. Wang, M. Amin, S. Lachgar, P. Mansfield, S. Prakash, B. Green, E. Dominowska, B. A. y Arcas, N. Tomasev, Y. Liu, R. Wong, C. Semturs, S. S. Mahdavi, J. Barral, D. Webster, G. S. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, and V. Natarajan, “Towards expert-level medical question answering with large language models,” 2023.

[3] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, p. 382–401, jul 1982. [Online]. Available: https://doi.org/10.1145/357172.357176

[4] F. B. Schneider, “Implementing fault-tolerant services using the state machine approach: a tutorial,” *ACM Comput. Surv.*, vol. 22, no. 4, p. 299–319, dec 1990. [Online]. Available: https://doi.org/10.1145/98163.98167

[5] R. Rodrigues, M. Castro, and B. Liskov, “Base: using abstraction to improve fault tolerance,” in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, ser. SOSP ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 15–28. [Online]. Available: https://doi.org/10.1145/502034.502037

[6] C. Cachin, S. Schubert, and M. Vukolic, “Non-Determinism in Byzantine Fault-Tolerant Replication,” in *20th International Conference on Principles of Distributed Systems (OPODIS 2016)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), P. Fatourou, E. Jiménez, and F. Pedone, Eds., vol. 70. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017, pp. 24:1–24:16. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2016.24

[7] C. Cachin, R. Guerraoui, and L. Rodrigues, *Introduction to Reliable and Secure Distributed Programming*, 2nd ed. Springer Publishing Company, Incorporated, 2011.

[8] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 3980–3990. [Online]. Available: https://doi.org/10.18653/v1/D19-1410

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *North American Chapter of the Association for Computational Linguistics*, 2019.

[10] M. P. Herlihy and J. M. Wing, “Linearizability: a correctness condition for concurrent objects,” *ACM Trans. Program. Lang. Syst.*, vol. 12, no. 3, p. 463–492, jul 1990. [Online]. Available: https://doi.org/10.1145/78969.78972

[11] C. Dwork, N. Lynch, and L. Stockmeyer, “Consensus in the presence of partial synchrony,” *J. ACM*, vol. 35, no. 2, p. 288–323, apr 1988. [Online]. Available: https://doi.org/10.1145/42282.42283

[12] M. Castro and B. Liskov, “Practical byzantine fault tolerance,” in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI ’99. USA: USENIX Association, 1999, p. 173–186.

[13] Y. Gao, Y. Lu, Z. Lu, Q. Tang, J. Xu, and Z. Zhang, “Dumbo-ng: Fast asynchronous bft consensus with throughput-oblivious latency,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1187–1201. [Online]. Available: https://doi.org/10.1145/3548606.3559379

[14] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias, “TRUE: Re-evaluating factual consistency evaluation,” in *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, S. Feng, H. Wan, C. Yuan, and H. Yu, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 161–175. [Online]. Available: https://aclanthology.org/2022.dialdoc-1.19

[15] T. D. Chandra and S. Toueg, “Unreliable failure detectors for reliable distributed systems,” *J. ACM*, vol. 43, no. 2, p. 225–267, mar 1996. [Online]. Available: https://doi.org/10.1145/226643.226647

Poster: Security and Privacy Heterogeneous Environment for Reproducible Experimentation (SPHERE)

Jelena Mirkovic*, David Balenson*, Brian Kocoloski*, David Choffnes†, Daniel Dubois†,
Geoff Lawler*, Chris Tran*, Joseph Barnes*, Yuri Pradkin*, Terry Benzel*,
Srivatsan Ravi*, Ganesh Sankaran*, Alba Regalado*, and Luis Garcia‡
* USC Information Sciences Institute, Email: mirkovic, balenson, bkocolos, glawler, ctran,
jdbarnes, yuri, tbenzel, sravi, sankara, alba@isi.edu
† Northeastern University, Email: choffnes@ccs.neu.edu, d.dubois@northeastern.edu
‡ University of Utah, Email: la.garcia@utah.edu

***Abstract*—To transform cybersecurity and privacy research into a highly integrated, community-wide effort, researchers need a common, rich, representative research infrastructure that meets the needs across all members of the research community, and facilitates reproducible science. USC Information Sciences Institute and Northeastern University are meeting researcher needs, and have been funded by the NSF mid-scale research infrastructure program to build Security and Privacy Heterogeneous Environment for Reproducible Experimentation (SPHERE). SPHERE research infrastructure will offer access to an unprecedented variety of user-configurable hardware, software, and network resources, it will offer six user portals geared toward different populations of users, and it will support reproducible research via a combination of infrastructure services and community engagement activities.**

I. Introduction

Cybersecurity and privacy threats increasingly impact our daily lives, our national infrastructures, and our industry. Recent newsworthy attacks targeted nationally important infrastructure, our government, our nuclear facilities, our researchers, and research facilities. The landscape of what needs to be protected and from what threats is continuously evolving: new technologies are released and the threat actors improve their own capabilities through experience and close collaboration. Meanwhile, defenders often work in isolation, using private data and facilities, and producing defenses that are quickly outpaced by new threats. To transform cybersecurity and privacy research into a highly integrated, community-wide effort, researchers need a common, rich, representative research infrastructure that meets the needs across all members of the research community, and facilitates reproducible science.

To meet researcher needs, USC Information Sciences Institute and Northeastern University have been funded by the NSF mid-scale research infrastructure program to build Security and Privacy Heterogeneous Environment for Reproducible Experimentation (SPHERE). This research infrastructure will offer access to an unprecedented variety of hardware, software, and other resources, all relevant to cybersecurity and privacy research, connected by user-configurable network substrate, and protected by a set of security policies uniquely aligned with cybersecurity and privacy research needs. SPHERE will offer six user portals, closely aligned with needs of different user groups, facilitating widespread adoption. It will provide built-in support for reproducibility, via easy experiment packaging, sharing, and reuse. SPHERE will build a process, a standard, and incentives for community-wide efforts to develop representative experimentation environments for cybersecurity and privacy research, and to continuously contribute high-quality research artifacts. You can learn more about SPHERE by visiting https://sphere-project.net.

II. Community Need

Over the past decade, and especially during the Covid-19 pandemic, both an individual's and society's essential functions (e.g., work, school, entertainment, social, financial, infrastructure, and governance) moved increasingly online. This sharply increased our nation's dependence on correct and reliable functioning of network and computing systems, and has led to increases in the frequency and impact of cybersecurity and privacy CS&P attacks. Recent years have seen unprecedented and record-breaking attacks, for example the Solar Winds supply-chain attack [5], which exposed confidential government data, and the Colonial Pipeline attack [8], which shut down our major gas pipeline for several days. Ransomware attacks more than tripled [6], DDoS attacks doubled [2], and data breaches increased by 70% [7]. Simply put, we now live in a world where cybersecurity and privacy are intrinsically intertwined with everything we do, and failures in these domains can have far-reaching monetary and national security impacts, and even jeopardize human lives. **Research progress in cybersecurity and privacy is thus of critical national importance**, to ensure safety of U.S. people, infrastructure and data.

USC Information Sciences Institute ran two workshops in 2022 to learn about community need around cybersecurity and privacy research: the *Cybersecurity Artifacts Workshop* [1] and the *Cybersecurity Experimentation of the Future 2022 Workshop* [4].

CS&P researchers need **common, rich, representative research infrastructure, which meets the needs across all members of the community, and facilitates reproducible science** to move from *piecemeal, opportunistic research* to *pursuing integrated, sophisticated, community-encompassing research.*

III. SPHERE Research Infrastructure

We are building innovative, transformative research infrastructure (RI) for CS&P experimentation: SPHERE – Security and Privacy Heterogeneous Environment for Reproducible Experimentation. In this section we describe the architecture, services, and community-building activities we plan to undertake to transform CS&P research from piecemeal and opportunistic to highly integrated, community-wide effort that is sophisticated and reproducible.

The SPHERE research infrastructure will offer rich, abundant, and diverse hardware resources, which would meet the experimental needs of 90% of researchers today [3]. The devices we plan to purchase and integrate with SPHERE as *experimental nodes*, and the research that benefits from these are as follows: (1) **General compute nodes:** 48 from DeterLab, 144 new nodes, with Intel TDX, ARM CCA/TrustZone, and AMD SEV; **Research supported:** application, system and network security, measurement, human user studies, large-scale experiments, education, trustworthy computing; (2) **Machine learning nodes:** 10 GPU-equipped servers; **Research supported:** security with machine-learning in the loop; (3) **Cyber-physical nodes:** 15 Rockwell Automation ControlLogix PLCs, I/O modules; **Research supported:** critical infrastructure security; (4) **Embedded compute nodes:** 600 from DCOMP, 312 new (Intel Atom, Intel Xeon D, ARM Cortex-A57, and NVIDIA Jetson NX Volta GPUs); **Research supported:** edge computing security, blockchain security, private computing, trustworthy edge computing, federated learning; (5) **IoT nodes:** 500 IoT nodes (a variety of smart home, smart speaker, camera, doorbell, TV, appliance, medical, office, wearable, and miscellaneous devices); **Research supported:** IoT security, user privacy; and (6) **Programmable nodes:** 8 programmable switches, 16 NetFPGA development boards (smartNICs); **Research supported:** dynamic (programmable) network security, SDN security. SPHERE will support most popular and relevant devices for CS&P research today. If CS&P research trends change in the future, new devices can be easily added by adding new installation and control scripts.

Many CS&P researchers study phenomena that interact closely with network topology, protocols and actors – SPHERE will meet the field's unique needs by offering a dedicated, user-configurable network substrate. CS&P experiments further may include generation of harmful traffic, taking live measurements from the real Internet, running human user studies, and even interacting with malicious Internet actors. To support these different research needs, and protect the Internet, SPHERE will provide safe network security policies.

All SPHERE nodes will be accessible via a single user interface. To meet the needs of various classes of users, SPHERE will provide six user portals: MAN (manual) - for exploratory research, JUP (Jupyter) – for mature research, GUI – for novice users, EDU – for use in education, AEC – for artifact evaluation committees, and HUM – for human user studies. Users will be able to access all portals from the user interface, and obtain a consistent view of their experiments, while being able to switch between portals as their needs evolve.

SPHERE will promote integrated research in cybersecurity and privacy and facilitate reproducible science by building a streamlined process, standards, and incentives for the community to develop, share and reuse high-quality research artifacts. To aid artifact packaging, SPHERE will build infrastructure services that include extensive logging of user actions and support for various approaches to capture experiment topology, setup and workflow. In addition to these technological advances, SPHERE team will engage with artifact evaluation committees at conferences and journals to support artifact evaluation on SPHERE. Additionally, SPHERE will issue an open call for mature research artifacts to be deployed on SPHERE as representative experimentation environments.

IV. Conclusion

This poster describes SPHERE[1], a new research infrastructure for cybersecurity and privacy that will be built over the next four years by USC-ISI and Northeastern University. It is our hope that SPHERE will transform and propel CS&P research to new advances, by providing a common experimentation platform for the research community.

References

[1] D. Balenson, J. Mirkovic, E. Eide, L. Tinnel, T. Benzel, D. Emmerich, and D. Johnson, "Cybersecurity artifacts workshop – report," https://bit.ly/CyberArtifactsWkshp2022, 2022.

[2] Government Technology, "Hacktivism and DDOS Attacks Rise Dramatically in 2022," https://www.govtech.com/blogs/lohrmann-on-cybersecurity/hacktivism-and-ddos-attacks-rise-dramatically-in-2022.

[3] J. Mirkovic, "Survey of Experimentation Approaches in Cybersecurity and Privacy Papers," https://bit.ly/CyberPapersSurvey2022, 2022.

[4] J. Mirkovic, D. Balenson, S. Ravi, L. Garcia, and T. Benzel, "Cybersecurity Experimentation Workshop – 2022 – Report," https://bit.ly/CyberExperWkshp2022, 2022.

[5] NPR, "A 'Worst Nightmare' Cyberattack: The Untold Story Of The SolarWinds Hack," https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack.

[6] Statista, "Annual number of ransomware attacks worldwide from 2016 to first half 2022," https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/.

[7] Sumeet Wadhwani, Spiceworks, "Data Breaches Soared by 70% In Q3 2022 in an Otherwise Dull Year," https://www.spiceworks.com/it-security/data-security/news/data-breach-report/, 2022.

[8] TechTarget.com, "Colonial Pipeline hack explained: Everything you need to know," https://www.techtarget.com/whatis/feature/Colonial-Pipeline-hack-explained-Everything-you-need-to-know.

[1] SPHERE is based upon work supported by the National Science Foundation under award number 2330066. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Poster: Improved Federated Learning with Non-IID Data Using Foundation Models

Fatima Abacha*, Sin G. Teo†, Lucas C. Cordeiro* and Mustafa A. Mustafa*‡
*Department of Computer Science, The University of Manchester, Manchester, M13 9PL, UK
†Institute for Infocomm Research, A*STAR Singapore
‡COSIC, KU Leuven, Leuven, 3001, Belgium
Email: fatima.abacha@postgrad.manchester.ac.uk

***Abstract*—Federated Learning (FL) enables multiple parties to train a model without sharing data. However, in heterogeneous scenarios where the data distribution amongst the FL participants is non-independent and identically distributed (non-IID), FL suffers from the data heterogeneity challenge which severely degrades the ability of the global model to converge. To solve this problem, we propose a novel data augmentation strategy, named DPSDA-FL, which can aid in homogenizing the local data present on the client's side. DPSDA-FL improves the training of the global model by leveraging differentially private synthetic data from foundation models. We obtain promising preliminary results on the CIFAR-10 dataset regarding recall of the global model.**

1. Introduction

Federated Learning (FL) enables multiple parties to come together and train an ML model without sharing their data [1]. The training process is orchestrated by a third party, which is usually a central server. In FL, each client uses private data to train its own local model, while the server uses an aggregation algorithm to construct a global model. The entire process runs for several iterations until a global model with the desired performance is achieved.This global model is then broadcast to all clients so they can use it for inference on their test dataset. FL protects against data leakage as each client's private training data is not disclosed to any other party. FL also enables adherence to regulations such as the GDPR [2].

However, when the data distribution amongst the clients in FL is statistically heterogeneous, meaning the data distribution is non-independent and identically distributed (non-IID), the prediction accuracy of the models is affected. A client may hold data from some classes and not from other classes present in the global dataset, as such the ability of the global model to make accurate inferences is severely degraded [3]. Also, when clients train their local model on data that does not contain certain classes from the global set or only a few samples from specific classes, the models will likely be biased towards those underrepresented groups [3]. This could have devastating consequences when these models are deployed in safety-critical situations such as healthcare and finance.

Statistical heterogeneity can be tackled by making the dataset across FL clients uniform in their distribution. This can be achieved by using data augmentation, a technique that can generate more training data to harmonize the data distribution amongst the clients in FL. Data augmentation is effective as it can reduce the problem of non-IID data in FL [3]. Existing works [4], [5] in the literature have used Generative Adversarial networks (GANs) to generate synthetic data for data augmentation. GANs, however, are vulnerable and tricky to train to produce high-quality and diverse synthetic data for data augmentation [6]. In this regard, we propose using foundation models for a more effective data augmentation process. To the best of our knowledge, this is the first work that employs pre-trained foundation models to generate differentially private synthetic data to tackle the problem of non-IID Data in FL. Thus, our contributions are as follows:

- We propose a new data augmentation strategy, named Differentially Private Synthetic Data Aided Federated Learning using Foundation Models (DPSDA-FL), to enhance the FL performance with non-IID Data.
- We demonstrate the effectiveness of utilizing Differentially Private Synthetic Data from Foundation Models in Cross Silo Horizontal FL.
- We conduct experiments and evaluations on the CIFAR-10 dataset and obtain enhancements regarding the recall capability of the global model.

2. Related Work

2.1. Data Heterogeneity and Data Augmentation

Data Heterogeneity in FL arises from the differences in data distribution and quantity among participants. Quantity skew results from the differences in the amount of data held by clients, while label skew – the differences in the classes of data held by individual clients [7].

Several techniques have been proposed to address the challenge of data heterogeneity in FL. FedProx [8] integrates a proximal term into the training process, which reduces the divergence of the local models from the global model by serving as a penalization term. In [9], a stochastic controlled averaging algorithm, a modification of the federated averaging, was developed, which incorporates variance reduction to stabilize the local model towards the global model. However, these techniques are not effective in extreme cases of data heterogeneity.

Another line of work uses GANs to mitigate the effects of data heterogeneity by generating additional training data for data augmentation [10]. In [4], a GAN was trained at the server side using FL and then used to generate synthetic data. This data is shared across clients to improve FL's performance. Other researchers proposed Synthetic Data Aided Federated Learning (SDA-FL) [5], where all clients receive a portion of locally synthetically generated data that is globally shared by the server. Despite the effectiveness of GAN-based methods in combating data heterogeneity problems in FL and enhancing the performance of the global model, these works have limitations. The instability of training GANs can result in low-quality synthetic samples with low utility [11].

Recent works have addressed the underperformance of GANs in generating high-quality synthetic data by adopting diffusion models. These models have been shown to produce high-quality data for computer vision applications [6], [11]. Diffusion models, however, can be challenging to train due to their high computational requirements. However, the emergence of foundation models has made access to pre-trained diffusion models more accessible. Foundation models like Open AI's Stable Diffusion and DALL.E [12] have become widely accessible to the public. These pre-trained models can be leveraged to generate high-utility synthetic data.

2.2. Differentially Private Synthetic Data Using Foundation Models

Synthetic data has been demonstrated to inadvertently reveal sensitive information about the original dataset generated from [13]. Consequently, integrating privacy-preserving techniques into the synthetic data generation process is imperative. Differential Privacy (DP) is a method that introduces randomness while computing statistics to maintain the privacy of the underlying information [14]. It has emerged as the standard approach for enhancing the privacy of synthetic data due to its ability to offer provable privacy guarantees. Consequently, diffusion models are being trained using DP to safeguard the privacy of the synthetic data they produce.

In [15], by fine-tuning pre-trained diffusion models with tens of millions of parameters, high utility data with low Fréchet Inception Distance (FID) were generated privately. The synthetic data was employed for a downstream classification task, and state-of-the-art results were attained. A more recent method, PRIVIMAGE [16], generates DP synthetic images using foundation models by strategically selecting pre-training data. While this approach is practical, it incurs significant memory and time overheads. Another notable technique is Private Evolution (PE) [17], an algorithm that fine-tunes pre-trained diffusion models to generate synthetic data from private datasets while maintaining differential privacy. PE has demonstrated state-of-the-art results in image synthesis and requires no pre-training. In this study, we leverage PE to generate synthetic data for data augmentation.

3. DPSDA-FL: Differentially Private Synthetic Data Aided Federated Learning Using Foundation Models

Algorithm 1 and Figure 1 give a high-level overview of our novel data augmentation strategy, DPSDA-FL, which works as follows:

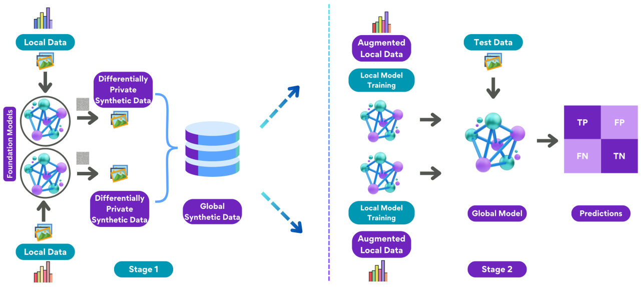

Figure 1. DPSDA-FL: Differentially Private Synthetic Data Aided Federated Learning Using Foundation Models.

Algorithm 1 DPSDA-FL

```
Input Parameters:
$N$: Number of clients.
$T$: Total number of rounds.
$\alpha$: Learning rate.
$w_t$: Initial model parameters.
$w_{t+1}$: Updated model parameters.
Initialization
Clients share their unique label counts with a server
Clients generate DP synthetic data using FMs
for $i = 1$ to $N$ do
    Generate $D^i_{\text{syn}}$ from $D^i_c$
    Send $D^i_{\text{syn}}$ to server
end for
Server forms global $D_{\text{Gsyn}}$ from $D^i_{\text{syn}}$
Distribute $D_{\text{Gsyn}}$ using unique label count
for $t = 1$ to $T$ do
    Send $w_t$ to all clients
    for $i = 1$ to $N$ do
        Augment $D^i_c$ with $D_{\text{Gsyn}}$
        Train model $L_i$ to update $w^i_{t+1}$
        Server Initializes $w_0$
        Send $w^i_{t+1}$ to server
    end for
    Aggregate $w_{t+1} = \frac{1}{N}\sum_{i=1}^{N} w^i_{t+1}$
end for
Repeat until convergence =0
```

1) At the start of the training process, clients share their unique label counts with the server to form a globally unique label count.
2) Each client proceeds to locally use the image-guided diffusion model locally [17] to generate differentially private synthetic data D_{syn} from their local data.
3) The diverse and high-quality generated synthetic data are then shared with the server to form a global synthetic data D_{Gsyn}.
4) The server uses the unique label count to distribute the synthetic data to the clients.
5) Each client augments its local data with the global synthetic data to homogenize its local data and train its local model.
6) The training process proceeds as in FedAvg for multiple rounds until the global model converges.

TABLE 1. EXPERIMENTAL SETTINGS

Name	Value
FL architecture	Cross-Silo Horizontal FL
Dataset	CIFAR-10
NN architecture	CNN
Number of clients	5
Non-IID partition	2 classes per client
Number of local epochs	2
Number of global rounds	20
Learning rate	0.1
Batch size	32
Optimizer	Stochastic gradient descent

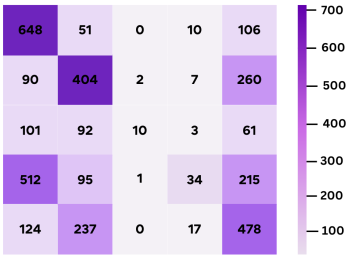

Figure 2. Confusion matrix for the baseline approach: FedAvg.

4. Experiments and Preliminary Results

Our experimental settings and results are summarised in Table 1, and Fig. 2 and 3, respectively. As can be observed, the global model trained using DPSDA-FL results in more correct predictions when compared to the baseline: FedAvg. These results suggest that differentially private synthetic data generated by foundation models can be utilized for local model training to mitigate the effects of data heterogeneity in FL with Non-IID Data.

5. Conclusions and Future Work

Our work presents a novel data augmentation technique for cross-silo horizontal FL designed to address the non-IID Data challenge. This technique enables clients to generate high-quality, diverse, differentially private data, which can be shared to enhance local model training. The results of our experiments show a significant improvement of up to 27% in the recall of the global model trained with DPSDA-FL, compared to the baseline: FedAvg. This underscores the potential of our technique to significantly enhance the performance of global model training in FL.

As future work, we will experiment with datasets that do not overlap with the pretraining data of the foundation models.

Acknowledgements

This work was supported by The University of Manchester through their PhD scholarships and EPSRC through the EnnCore project [EP/T026995/1].

References

[1] H Brendan McMahan, Eider Moore, Daniel Ramage, and Seth Hampson. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2017.

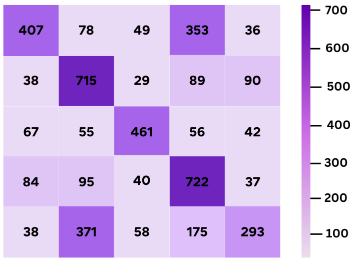

Figure 3. Confusion matrix for our proposed approach:DPSDA-FL.

[2] Chunyi Zhou, Anmin Fu, Shui Yu, Wei Yang, Huaqun Wang, and Yuqing Zhang. Privacy-Preserving Federated Learning in Fog Computing. *IEEE Internet of Things Journal*, 7(11):10782–10793, November 2020.

[3] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated Learning with Non-IID Data. 2018. arXiv:1806.00582 [cs, stat].

[4] Longling Zhang, Bochen Shen, Ahmed Barnawi, Shan Xi, Neeraj Kumar, and Yi Wu. FedDPGAN: Federated Differentially Private Generative Adversarial Networks Framework for the Detection of COVID-19 Pneumonia, April 2021. arXiv:2104.12581 [cs, eess].

[5] Zijian Li, Jiawei Shao, Yuyi Mao, Jessie Hui Wang, and Jun Zhang. Federated Learning with GAN-based Data Synthesis for Non-IID Clients, June 2022. arXiv:2206.05507 [cs].

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

[7] Liangqiong Qu, Niranjan Balachandar, and Daniel L. Rubin. An Experimental Study of Data Heterogeneity in Federated Learning Methods for Medical Imaging, July 2021. arXiv:2107.08371 [cs].

[8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks, April 2020. arXiv:1812.06127 [cs, stat].

[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning, April 2021. arXiv:1910.06378 [cs, math, stat].

[10] Roozbeh Razavi-Far, Ariel Ruiz-Garcia, Vasile Palade, and Juergen Schmidhuber, editors. *Generative Adversarial Learning: Architectures and Applications*, volume 217 of *Intelligent Systems Reference Library*. Springer International Publishing, Cham, 2022.

[11] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic Data from Diffusion Models Improves ImageNet Classification, 2023. arXiv:2304.08466 [cs].

[12] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, April 2022. arXiv:2204.06125 [cs].

[13] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. A Unified Framework for Quantifying Privacy Risk in Synthetic Data, November 2022. arXiv:2211.10459 [cs].

[14] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[15] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L. Smith, Olivia Wiles, and Borja Balle. Differentially Private Diffusion Models Generate Useful Synthetic Images, February 2023. arXiv:2302.13861 [cs, stat].

[16] Kecen Li, Chen Gong, Zhixiang Li, Yuzhong Zhao, Xinwen Hou, and Tianhao Wang. PrivImage: Differentially Private Synthetic Image Generation using Diffusion Models with Semantic-Aware Pretraining, April 2024. arXiv:2311.12850 [cs].

[17] Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially Private Synthetic Data via Foundation Model APIs 1: Images, 2024. arXiv:2305.15560 [cs].

Poster: Integrating a Secure Processing Environment in an IoT Operating System

Lena Boeckmann
HAW Hamburg
Hamburg, Germany
lena.boeckmann@haw-hamburg.de

Thomas C. Schmidt
HAW Hamburg
Hamburg, Germany
t.schmidt@haw-hamburg.de

Matthias Wählisch
TU Dresden
Dresden, Germany
m.waehlisch@tu-dresden.de

***Abstract*—Trusted Execution Environments (TEE) and secure enclaves with hardware support are promising concepts for enhancing security in constrained environments. These approaches provide protected processing areas within a SOC, in which security-critical applications can execute, and at the same time prevent unauthorized access to sensitive data and program code. New microcontrollers with the Armv8-M architecture offer Trustzone-M, a hardware feature to protect memory and support TEEs. To facilitate adoption, Arm provides an open source reference implementation for a secure processing environment (Trusted Firmware-M). In this poster, we present how we integrated this secure firmware in an IoT operating system and measure the overhead cost in memory and execution time.**

***Index Terms*—Internet of Things, IoT, Security, Trusted Execution Environment, Trustzone-M**

I. Introduction

Internet of Things (IoT) devices store, process and transmit sensitive data, while often being insecure and easily physically accessible by potential attackers. Vulnerable devices can serve as entry points to larger networks for compromising critical system components and infrastructure. To secure IoT systems we need measures to make those devices trustworthy.

One way to achieve this are Trusted Execution Environments (TEE) [1]. Those are isolated components in which trusted applications (TA) can perform security critical operations, such as secure storage of data and cryptographic key material, cryptographic operations, device authentication and attestation and secure over-the-air (OTA) updates.

TEEs can provide a reduced set of operations only required to establish trust between communication partners and expose a smaller attack surface than a rich OS. An OS could be compromised by malware or through a physical attack. Separating critical operations from the OS provides an extra layer of security and allows for independent attestation and verification.

In the constrained IoT, hardware-supported TEEs can help to protect devices. On Arm Cortex-M devices with the Armv8-M and Armv8.1-M architectures, TrustZone-M allows for a memory-map based system separation [2]. Flash and memory are split into secure and non-secure address regions, allowing access to secure addresses only, when the CPU runs in a secure state. CPU state transitions are performed in hardware, either by triggering interrupts or through non-secure callable veneer functions, aiming to make them fast and efficient.

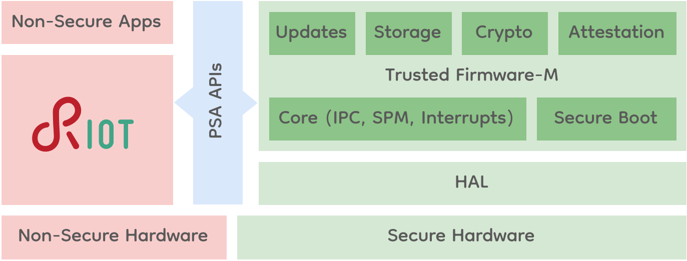

Fig. 1. TF-M in combination with a non-secure operating system like RIOT

To increase the security of the IoT operating system RIOT [5], we aim to integrate a secure firmware with the OS. RIOT is an open source project aiming at a small memory footprint and support for many different architectures. This is achieved by a minimalistic core, which can be extended with optional feature modules.

One existing candidate for a secure firmware on Arm Cortex-M platforms, is the open source project Trusted Firmware-M (TF-M) [4]. TF-M is a reference implementation of a *Secure Processing Environment (SPE)* [3], which has been specified as part of the Arm Platform Security Architecture (PSA) framework. Since RIOT already supports the PSA Crypto API [6], we decided to also evaluate the suitability of TF-M as a secure firmware in RIOT.

In this poster, we report on ongoing work to leverage TEE technologies in RIOT. We partly integrated TF-M with the OS and document the steps needed to run RIOT side by side with the firmware (§ II). We then measure the overhead introduced by the secure firmware (§ III), and describe the problems and limitations we encountered (§ IV).

II. Integrating TF-M in RIOT

TF-M provides a SPE, which acts as an intermediary between a *non-secure processing environment (NSPE)* and the secure hardware. It provides secure services, which are necessary to verify system integrity and increase security. Those services include secure updates, cryptography, secure storage (e.g. for keys and certificates) and attestation. The NSPE can be either a bare-metal application or an operating system, and runs in non-secure memory areas. Communication

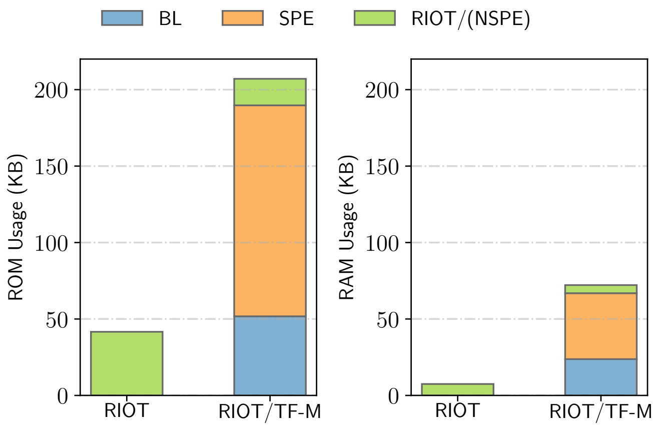

Fig. 2. RAM and ROM usage of RIOT and RIOT/TF-M builds

between SPE and NSPE happens through several service APIs, which are provided by the SPE.

The TF-M build system generates three binaries: a bootloader, a secure firmware image and, optionally, a non-secure firmware image. All images are signed with a key to be verified by the bootloader and concatenated to a single binary. When flashing, at first the bootloader is written at memory address 0, followed by the merged binary. If the image signatures are valid, the bootloader boots the SPE, which sets up the platform. The setup includes the configuration of secure memory regions and the creation of partitions for the secure services. All of this is done in a secure CPU state. The SPE then loads and boots the NSPE and triggers a transition to non-secure mode. If the non-secure application calls the secure service APIs, they trigger the transition back to the secure state to perform the requested operations. Afterwards the system switches back to non-secure mode and returns the results of the operation to the NSPE.

For this work, we aim to build RIOT as a NSPE and link it with TF-M. The TF-M documentation lists a few requirements for an OS to run as a NSPE:

- The OS must be able to run in non-secure mode.
- The OS must initialize the PSPLIM register and handle it during thread context switch operations.
- The OS needs to ensure that a link register value can differentiated between secure and non-secure builds.

First we had to figure out the details on what this meant for RIOT.

A. Preparing RIOT

Two of the platforms supported by TF-M are also supported in RIOT: the Nordic nRF9160 and nRF5340. There is a crypto hardware driver available for the nRF9160, providing more possibilities for experimentation, which is why we focus on this board for now. Per default, RIOT runs in secure mode and has access to the whole system, including secure RAM, peripherals and registers. To enable non-secure mode, we needed to find all the instances where secure addresses are accessed and change them to non-secure addresses. To make this optional, we added the modification as a compile time option, as shown in Listing 1.

```
#if IS_ACTIVE(MODULE_TRUSTED_FIRMWARE_M)
    #define LED_PORT            (NRF_P0_NS)
#else
    #define LED_PORT            (NRF_P0_S)
#endif
```

Listing 1. Example of optional access to secure and non-secure LED ports in RIOT

To switch between secure and non-secure images, supervisor calls (SVC) are needed. Per default, RIOT does not use SVC, so they have to be enabled explicitly when used with TF-M.

TF-M requires a secure RAM range of 0x16000 bytes, which means that RIOT RAM can only start at address 0x20016000. Usually RIOT RAM starts at 0x20000000, so we needed to modify RIOT RAM length and start address. This can be configured individually for RIOT platforms. Since TF-M uses MCUboot we could use RIOTs existing partial support for MCUboot to facilitate the integration. MCUboot usage requires the definition of the image header size and the new start address of the binary. As with the modified RAM start address, this can be configured in the CPU specific makefiles in RIOT.

Per default RIOT can only flash one binary. We added a TF-M specific makefile to RIOT that contains a new flash target with support for multiple binaries. This makefile is executed after building the secure and non-secure images. It links both images, signs them separately with an RSA key and merges them into one binary. It then flashes the bootloader binary at address 0x00 and the merged binary with the required offset.

B. Adding TF-M

Trusted Firmware-M has been added to RIOT as a third-party package. We implemented an interface, through which non-secure calls from the non-secure to the secure side can be executed. A makefile downloads the source code and builds TF-M in two steps. First, the secure image is configured and built. This produces the bootloader binary and secure binary, as well as a folder called *api_ns*. This contains code and configurations that are needed for communication between the secure image and the non-secure image. In the next step, we compile this *api_ns* and create a library that shall be linked with the RIOT binary.

C. Limitations

Per default, RIOT runs applications in a main thread, which is created during kernel initialization and provides its own stack with stacksize configured at runtime. After thread creation RIOT initiates a context switch to execute the program until completion. The current integration with TF-M does not permit RIOT to create its own threads for applications. This means, core threading needs to be disabled and OS and applications are run in the same thread. Since the required stack size for an application is defined when creating the main thread, there is currently no way to dynamically increase the

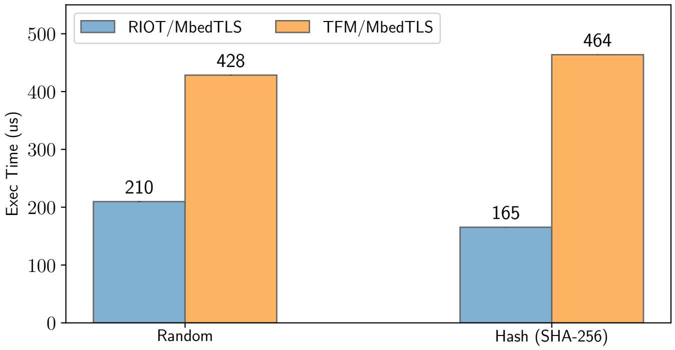

Fig. 3. Random and hash execution times by operation

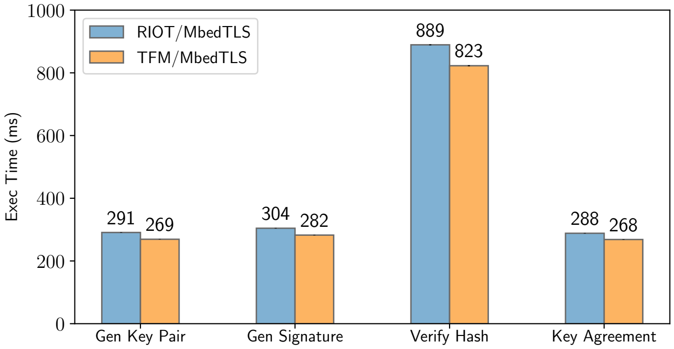

Fig. 4. ECC execution times by operation

stack size without threading. As a workaround, the stack size is hardcoded in the linker file for the nRF9160.

III. Evaluation

To evaluate the impact of TF-M on RIOT OS, we compare memory consumption and execution time of cryptographic operations with and without a secure processing environment. TF-M uses the PSA Crypto reference implementation from the MbedTLS library [7]. For comparison we include MbedTLS as a third-party package in RIOT. We use the same version TF-M uses and build it with the same configuration for the library and the PSA Crypto module. The only difference is that TF-M builds MbedTLS with the SPM (secure partition manager) option, to separate code into secure and non-secure parts. Since RIOT does not support this split we cannot use this option in our case.

A. Memory

Figure 2 shows the RAM and ROM usage of both builds. When building only RIOT with MbedTLS, we use $\approx 41\ KB$ of ROM and $\approx 7.5\ KB$ of RAM. When building RIOT as a NSPE for TF-M, the ROM usage decreases, because the MbedTLS library is now part of the SPE. The SPE now uses $\approx 140\ KB$ of ROM. Additionally the TF-M build adds $\approx 52\ KB$ of bootloader code. RIOT RAM usage also decreases slightly when building with TF-M, while the SPE and the bootloader use $\approx 43\ KB$ and $\approx 24\ KB$.

B. Execution Time

We measure the execution times of random number generation (RNG), a SHA-256 hash computation, elliptic curve (ECC) key generation, ECDSA sign/verify operation and an ECDH key agreement operation with a NIST-P256 curve. We measure each operation for the duration of 1000 iterations by toggling a GPIO before and after the execution. We sample the data with a logic analyzer at a rate of 6 MS/s. When comparing RNG and hash computation, TF-M introcudes a significant overhead of $\approx 200\%$ for RNG and $\approx 300\%$ for hash operations. Surprisingly, this observation can't be reproduced for elliptic curve operations. Here the TF-M build is around $20\ ms$ faster when generating a key pair, a signature and a shared secret, while being around $70\ ms$ faster when verifying a signed hash. Likely reasons are optimizations by TF-M as well as different `calloc` implementations. MbedTLS in TF-M allocates memory with a static buffer, while the RIOT version uses the slower libc implementation.

IV. Conclusions and Outlook

In this paper we integrated an existing implementation of a secure firmware in RIOT and measured the overhead it introduces to the operating system. We showed that the runtime overhead for random number and hash generation is quite large, while it is negligible for ECC operations. The firmware impact on RAM and ROM usage is quite large and does not comply with the RIOT goal of a small memory footprint. Also it is not portable to other CPU architectures. We therefore conclude that TF-M is not a desired solution for a secure firmware in RIOT. In future research, we will explore alternatives to TF-M and develop a new, RIOT specific TEE. The measurements we derived from this work will be used to compare our own implementation to TF-M and improve the size and efficiency of our design.

References

[1] "IEEE Standard for Secure Computing Based on Trusted Execution Environment," *IEEE Std 2952-2023*, pp. 1–29, 2023.

[2] S. Pinto and N. Santos, "Demystifying Arm TrustZone: A Comprehensive Survey," *ACM Comput. Surv.*, vol. 51, no. 6, jan 2019. [Online]. Available: https://doi.org/10.1145/3291047

[3] ARM Ltd., "Arm PSA Firmware Framework 1.0," https://developer.arm.com/documentation/den0063/a/?lang=en, last accessed 04-10-2024, 2019.

[4] ——, "ARM Trusted Firmware M," https://tf-m-user-guide.trustedfirmware.org, last accessed 04-30-2024, 2021.

[5] E. Baccelli, C. Gündogan, O. Hahm, P. Kietzmann, M. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch, "RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, December 2018. [Online]. Available: http://doi.org/10.1109/JIOT.2018.2815038

[6] L. Boeckmann, P. Kietzmann, L. Lanzieri, T. C. Schmidt, and M. Wählisch, "Usable Security for an IoT OS: Integrating the Zoo of Embedded Crypto Components Below a Common API," in *Proc. of Embedded Wireless Systems and Networks (EWSN'22)*. New York, USA: ACM, October 2022, pp. 84–95. [Online]. Available: https://dl.acm.org/doi/10.5555/3578948.3578956

[7] ARM Ltd., "Mbed TLS," https://tls.mbed.org, last accessed 07-17-2020, 2020.