Towards Lightweight Trusted Hardware-Assisted MPC

Wentao Dong[1], **Cong Wang**[1]

[1]*Department of Computer Science, City University of Hong Kong, KLN, Hong Kong SAR*

`wentdong-c@my.cityu.edu.hk, congwang@cityu.edu.hk`

1. Introduction

Secure multi-party computation (MPC) systems, which allow mutually distrusting parties to jointly evaluate a function without revealing input privacy, have prevailed in both academia and industry nowadays and predominantly fall into the following two categories:

- Cryptographic world: offering strong security albeit at high costs.
- Trusted hardware world: showing impressive performance but necessitating extra security assumptions.

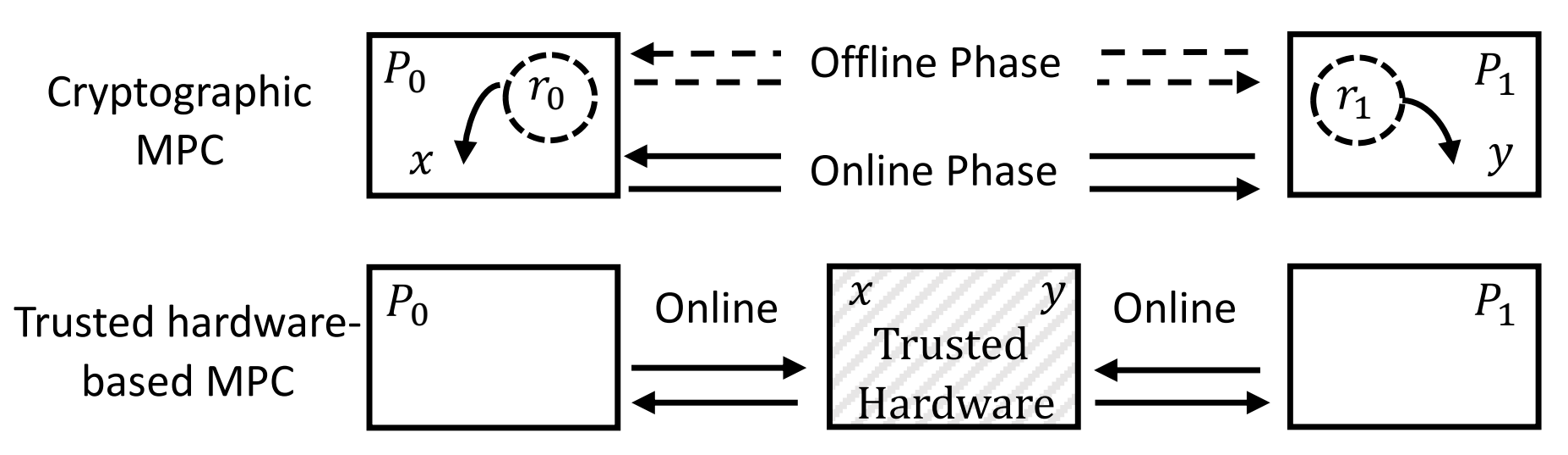


Figure 1: *Overview of cryptographic MPC in the preprocessing model and trusted hardware-based MPC.*

This work examines a natural problem of fusing the two worlds for both strong security and high performance by presenting a hybrid solution HPCG (short for ***hardware-assisted programmable pseudorandom correlation generator***). At a high level, HPCG is designed to facilitate the practical adoption of MPC techniques in the real world, relying only on lightweight trusted hardware (LTH) working offline (no need to access private inputs).

Main intuition: fusing them at a finer granularity.
Addressing the key efficiency bottlenecks of cryptographic MPC while reducing the use of and trust in the secure hardware itself.

2. Why Not Direct Trusted Hardware-Based MPC

How original trusted hardware-based MPC works:

- Employ high-performance trusted hardware like Intel SGX as a third-party black box and attest to its confidentiality and integrity.
- Load all sensitive data and code to perform secure computations on behalf of parties.

What is the main concern of original trusted hardware-based MPC:

- Availability of high-performance equipment like Intel SGX.
- Complexity in extensive legacy code refactoring and careful optimizations regarding speculation and caching for specific tasks
- Limitations of protected memory size.
- Increase in trusted computing base (TCB) and attack surfaces
- Potential risks of fully hosting sensitive data on trusted hardware.
- ...

A step forward – lightweight trusted hardware (LTH):
Different from high-performance equipment such as Intel SGX, is another flavor of secure hardware widely deployed nowadays. Informally, STH refers to small dedicated security processors like Google Titan and Apple T2 or on-chip security subsystems like Apple enclave processor.

- Much higher isolation level by being physically decoupled from the main processor (share no resources like memory and cache),
- Less complexity and smaller TCB
- More secure as shown by their successful history in high-security applications in the real world.
- Easy to deploy as a separate chip or an IP block.
- Albeit low performance.

An LTH-assisted MPC effort – Huang *et al.* [1]:
It employs LTH to compute expensive non-linear operations, while employing cryptographic tools to compute linear operations. It nicely reduces the TCB at the hardware level and improves online efficiency. However, it still

- Extra overhead due to IO between LTH and the main processor.
- Inherent risks of hosting inputs on trusted hardware.

3. Why Not Direct Cryptographic MPC

How cryptographic MPC works

- Parties first jointly execute an offline phase to generate and distribute some desired forms of input-independent correlated randomness (e.g., Beaver triple).
- Followed by a fast online phase that consumes these correlations and performs secure computations accordingly.

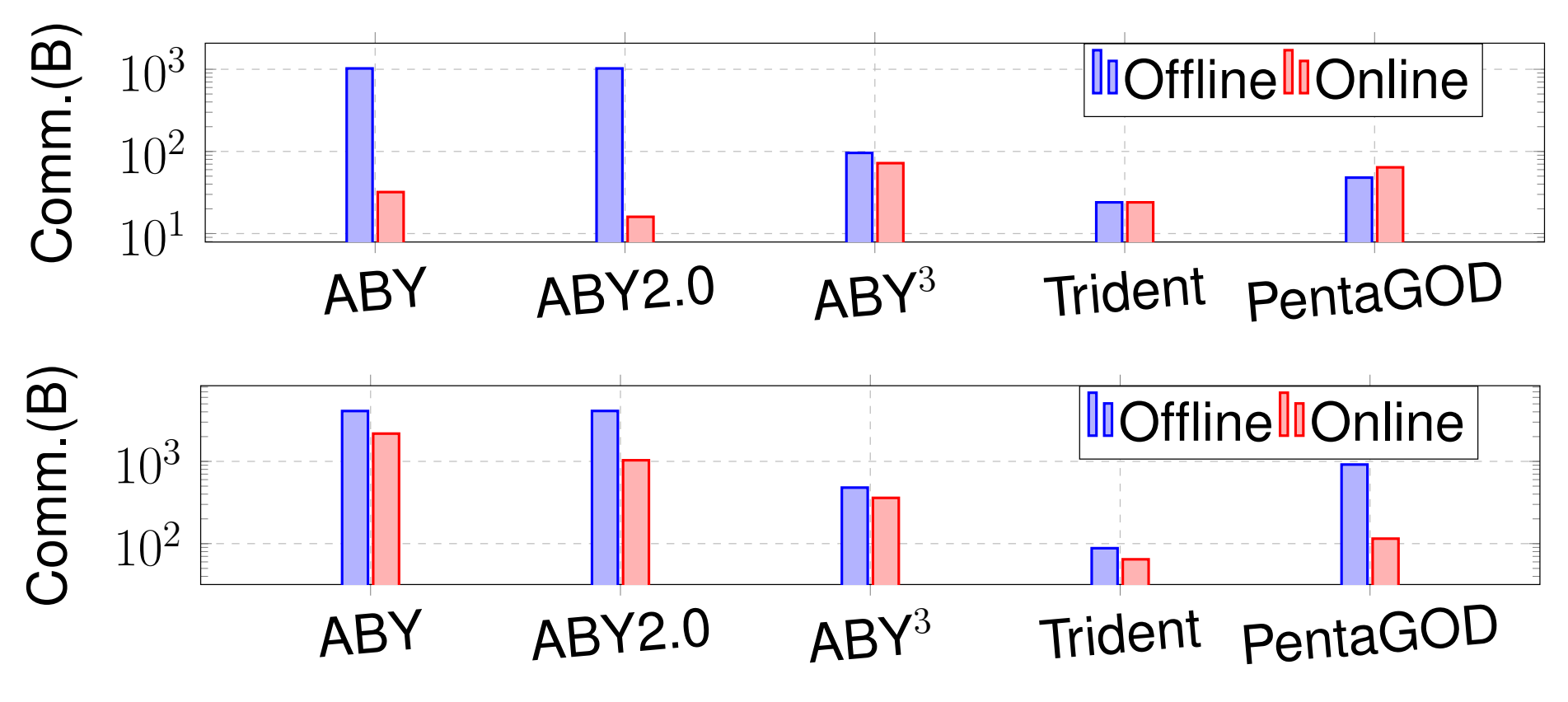


Figure 2: *Offline and online communications for secure multiplication and secure comparison in different MPC frameworks.*

What is the efficiency bottleneck of cryptographic MPC

- Offline overhead for generating and distributing useful correlated randomness forms an end-to-end bottleneck, as offline costs usually exceed online costs.
- Online costs of complicated functions form another end-to-end bottleneck, as they usually require more interaction rounds and incur long latency.

Without loss of generality, end-to-end efficiency is highly dominated by offline overhead and affected by more complex online computations, both attributed to correlated randomness (as shown in Fig. 2).

A step forward – silent preprocessing [2] by Boyle *et al.*
A PCG allows parties to stretch short correlated seeds into long sources of target correlations with sublinear communication and no interaction, aka *pseudorandom correlation generator* (PCG).

- Only available for some common forms of correlations (thus limited availability).
- Incur significant complexity to the system (thus limited usability).

A step forward – online optimized design trends [3]:
Many recent designs prioritize online efficiency metrics and shift as much input-independent computation as possible to offline (can be abstracted as complicated correlated randomness).

- Inevitably entail more offline penalties.
- Cannot work with current silent preprocessing designs.

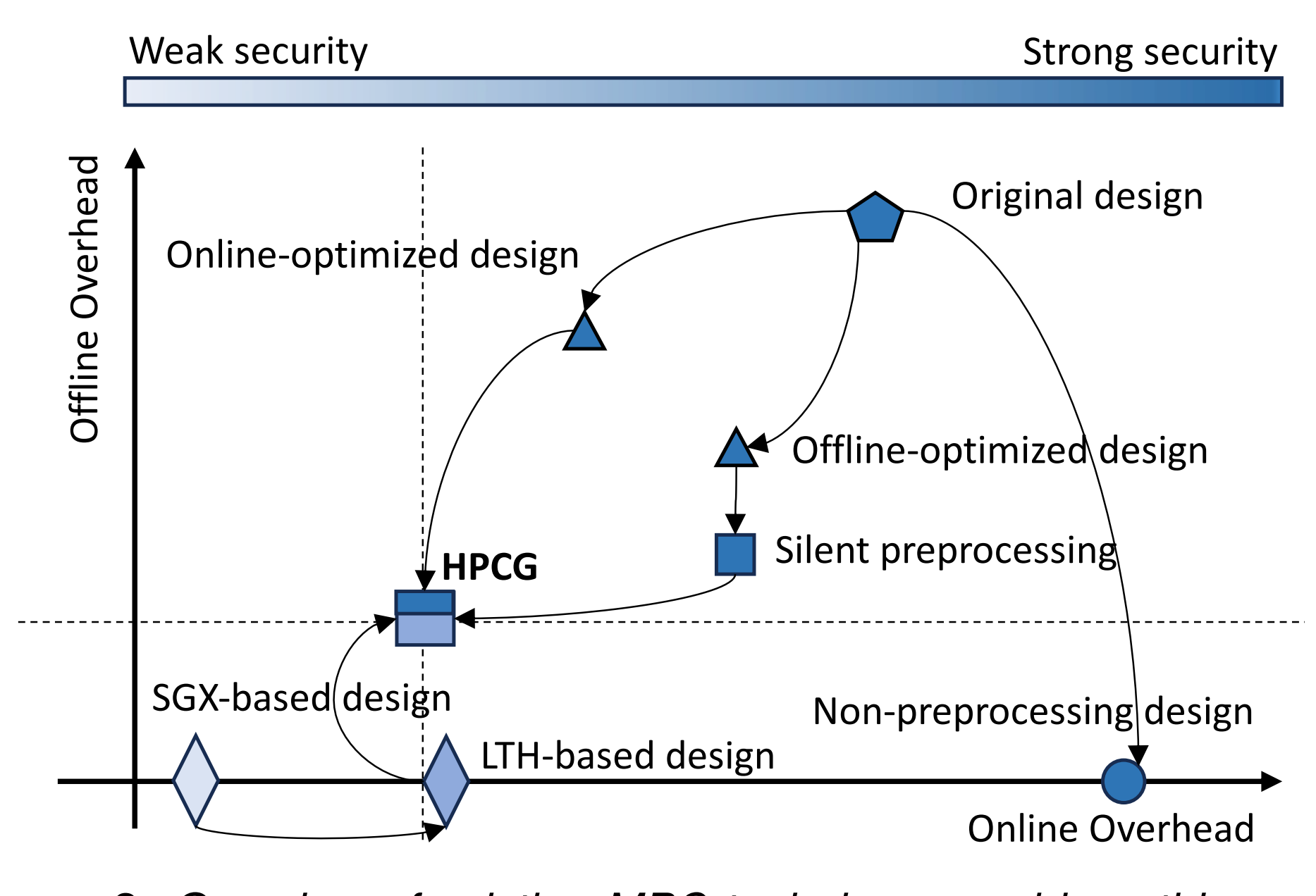
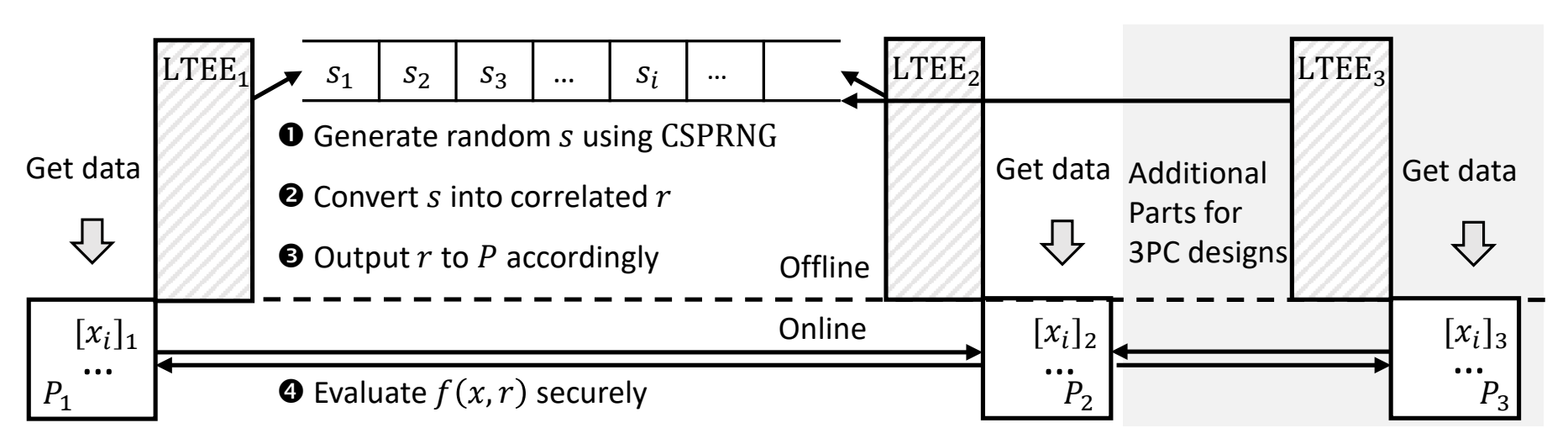


Figure 3: *Overview of existing MPC techniques and how this work is positioned in the design space.* HPCG *takes the starting point of silent preprocessing, STH-assisted design, and online-optimized works.*

4. Lightweight Trusted Hardware-Assisted Programmable Pseudorndom Correlation Generator

Putting recent advances from both hardware and MPC perspectives together, we get HPCG. Let $R_1, ..., R_n$ denote the target correlation shares required by $P_1, ..., P_n$, where these shares are correlated by some specifications $\mathcal{C}(R_1, ..., R_n) = 1$. Then an HPCG instance consists of a pair of functions $\mathsf{HPCG} = (\mathsf{Gen}, \mathsf{Expand})$ run in LTH such that:

- $\mathsf{Gen}(1^\lambda)$ is a interactive PPT algorithm that given security parameter λ, it generates a cryptographically-secure pseudorandom sequence $s = s_1, s_2, ...s_i, ...$.
- $\mathsf{Expand}(\mathsf{eid}, s, \mathsf{M})$ is a non-interactive PPT algorithm that on the LTH identifier eid and program M obtained during the setup, and the input s. LTHs invoke Sample and $\mathsf{RSample}$ on s to construct $(r_1, ..., r_n) \leftarrow s$ where $(r_1, ..., r_n) \in (R_1, ..., R_n)$ and output r_i to P_i accordingly.



LTEE_1
s_1
s_2
s_3
...
s_i
...
LTEE_2
LTEE_3
❶ Generate random s using CSPRNG
❷ Convert s into correlated r
❸ Output r to P accordingly
Get data
Get data
Additional Parts for 3PC designs
Get data
Offline
Online
$[x_i]_1$
P_1
$[x_i]_2$
P_2
$[x_i]_3$
P_3
❹ Evaluate $f(x, r)$ securely

Figure 4: *Overview of* HPCG, *it provides a generic template to convert cryptographically secure pseudorandomness into any required forms of correlated pseudorandomness and facilitate secure computations.*

Why HPCG?
HPCG functions by providing a generic, concretely efficient, and programmable PCG construction under small hardware trust only, making a rational compromise between hardware and cryptography worlds.

- Rely on LTH with more succinct TCB.
- Works in the offline phase (no need to access private inputs from users), leaving the online protocol security guarantee the same as cryptographic solutions.
- Improves both offline and online efficiency.

5. Implementation and Evaluation

We adopt the STM32F4 microcontroller (ARM Cortex-M4, 168 MHz) to instantiate HPCG The efficiency of HPCG is highly dominated by the throughput of CSPRNG and the complexity of target correlations (i.e., $|r|$, the number of consumed randomness per correlation). In practice, STM32F4 achieves 37.33 MB/s AES throughput, sufficient to construct many one-time correlation instances to facilitate online computations.

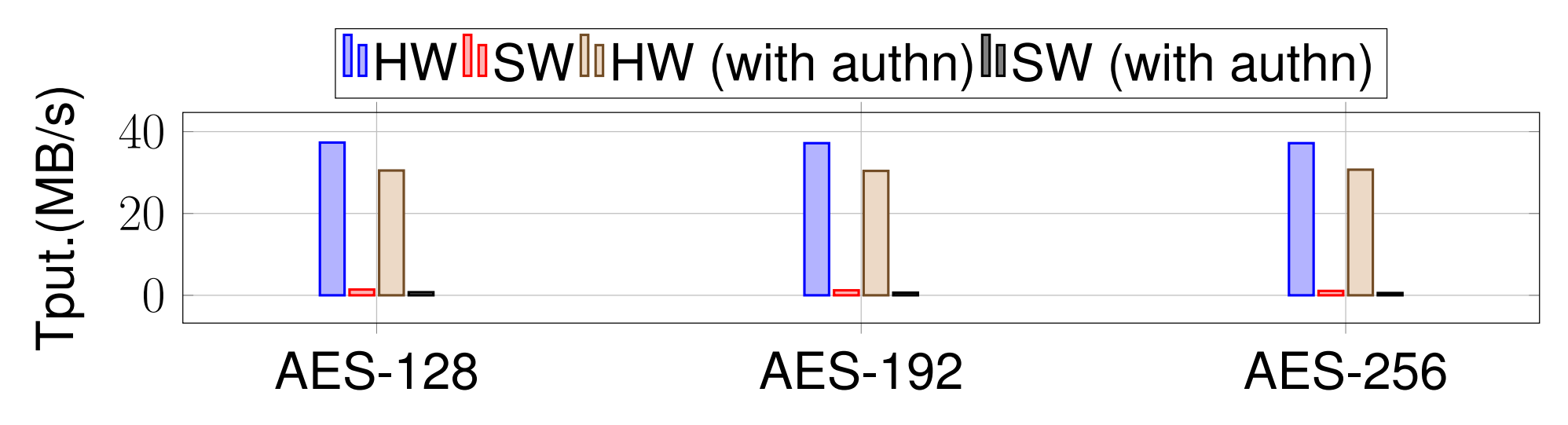


Figure 5: *Throughput of CSPRNG based on* AES-CTR, *HW/SW stands for hardware/software, authn stands for authentication*

Table 1: *Efficiency comparison for* HPCG *and prior works.*

Type	Corr.	Silent	Ref.	Comm.	Time
General-purpose	ABT (10^6)	No	Overdrive [4]	1.9 GB	16.91 s
		Yes	Boyle et al. [2] HPCG-ABT	2.77 MB **256 bit**	17.82 s **9.32 s**
	COT (10^7)	No	IKNP [5]	156 MB	15.53 s
		Yes	Boyle et al. [2] Silver [6] HPCG-COT	127 KB 127 KB **256 bit**	8.05 s 0.74 s **0.53 s**
	OLE (10^6)	No	Baum et al. [7]	482 MB	14.71 s
		Yes	HPCG-OLE	**256 bit**	**2.09 s**
Specialized	RPC (10^4)	No	RPM [8]	480 MB	5.74 s
		Yes	HPCG-RPC	**256 bit**	**2.61 s**
	EQC (10^6)	No	Baum et al. [7]	482 MB	15.18 s
		Yes	HPCG-EQC	**256 bit**	**2.51 s**

Table 2: *Efficiency comparison for 2PC circuit-based PSI.*

Ref.	Phase	2^{10}	2^{10}	2^{16}	2^{16}
		Comm.	Round/Time	Comm.	Round/Time
ABY [9]	Online	258 MB	50	-	-
ABY2.0 [3]	Online	53 MB	40	213 GB	64
HPCG	Offline Online	**256 bit** **8 MB**	**0.02 s** **0.36 s**	**256 bit** **32 GB**	**0.14 s** **524.41 s**

Table 3: *Efficiency comparison for 3PC secure shuffling*

Ref.	10^5	10^5	10^5	10^6	10^6	10^6
	Comm.	Round	Time	Comm.	Round	Time
Ruffle [10] HPCG	9.16 MB **1.61 MB**	2 **1**	0.46 s **0.46 s**	91.55 MB **16.04 MB**	2 **1**	7.03 s **5.86 s**

Our evaluations show that HPCG can concretely augment the efficiency and simplicity of MPC systems.

6. Concluding Remarks

Technical insights

- HPCG provides an efficient and generic approach that substantially extends the known feasible results of silent preprocessing to any required forms of correlated randomness (regardless of how complex their forms are).
- HPCG provides heuristics that can achieve better online simplicity and efficiency (e.g., optimal online round complexity) by enabling new forms of complicated target correlations for specific secure computation tasks, which can be of broad independent interests.

Use cases

- *Offline augmentation:* Improve legacy (existing) MPC systems by enabling silent preprocessing for better deployment and end-to-end efficiency. For instance, we can construct concretely efficient HPCG instances for more complicated correlations like daBits, edaBits, and oblivious punctured vectors.
- *Online augmentation:* Enable new MPC protocols by providing a powerful tool that facilitates the generation of new offline correlated randomness intentionally designed for specific tasks (although its design still requires additional efforts). For instance, the RPC for secure shuffling tasks and EQC for secure equality test form illustrative examples.

Independent interests

- *Malicious adversary.* HPCG can greatly avoid potential bandwidth wastage in cases where malicious adversaries intentionally generate malformed correlated randomness.
- *Traffic analysis* HPCG can mitigate traffic analysis attacks that anticipate future MPC plans.
- *Ad-hoc demand* HPCG is well-suited for ad-hoc or on-demand MPC cases, i.e., parties do not need to store massive target correlations in advance. They can quickly generate the required correlations.

References

[1] P. Huang, T. Hoang, Y. Li, E. Shi, and G. E. Suh, "Stamp: Lightweight tee-assisted mpc for efficient privacy-preserving machine learning," *arXiv prePrint Archive*, 2022.
[2] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl, "Efficient pseudorandom correlation generators: Silent ot extension and more," in *Proc. of CRYPTO*, 2019.
[3] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "$ABY2.0$: Improved mixed-protocol secure two-party computation," in *Proc. of USENIX Security*, 2021.
[4] M. Keller, V. Pastro, and D. Rotaru, "Overdrive: Making spdz great again," in *Proc. of EUROCRYPT*, 2018.
[5] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Proc. of CRYPTO*, 2003.
[6] G. Couteau, P. Rindal, and S. Raghuraman, "Silver: Silent vole and oblivious transfer from hardness of decoding structured ldpc codes," in *Proc. of CRYPTO*, 2021.
[7] F. Kerschbaum, E.-O. Blass, and R. A. Mahdavi, "Faster secure comparisons with offline phase for efficient private set intersection," in *Proc. of NDSS*, 2023.
[8] D. Lu and A. Kate, "Rpm: Robust anonymity at scale," in *Proc. of PETs*, 2023.
[9] D. Demmler, T. Schneider, and M. Zohner, "A framework for efficient mixed-protocol secure two-party computation," in *Proc. of NDSS*, 2015.
[10] N. Koti, V. B. Kukkala, A. Patra, B. R. Gopal, S. Sangal, *et al.*, "Ruffle: Rapid 3-party shuffle protocols," in *Proc. of PETs*, 2023.

Presented at the "ACM CCS", Copenhagen, Denmark, 2023