Poster: Fast Evaluation of S-boxes in MPC

Erik Pohle, Aysajan Abidin, Bart Preneel
imec-COSIC, KU Leuven, Belgium
firstname.lastname@esat.kuleuven.be

***Abstract*—Secure data collection from IoT devices for privacy-preserving data processing is an important part in data-driven and privacy-friendly business use-cases. One approach to ensure end-to-end confidentiality is symmetric encryption of the data. The encryption key can then be secret-shared among the data processors in advance. Once data arrive, the processors first compute the decryption in a distributed manner and then the desired data processing function using secure multi-party computation (MPC). Data processing results are similarly encrypted in a distributed manner and returned. In this poster we present work in progress on efficient symmetric encryption in two-party secure computation. Particularly, we present a new garbling scheme for Yao's garbled circuits that enables fast evaluation of S-boxes and SPN ciphers with certain properties. Further, we give performance estimates for selected (lightweight) symmetric primitives and preliminary results of an implementation of SKINNY.**

I. Introduction

The immense growth of Internet-connected devices, such as smartphones, smartwatches, wearables, medical implants, thermostats, and the city-wide deployments of IoT sensors present both opportunities and challenges. The data collected by these devices allow, for example, businesses to develop and offer personalized services and applications. There are, however, barriers due to the associated security and privacy risks, including, among others, secure data collection and processing. One way to achieve secure data collection is to encrypt the data and store it in a centralized database. Storing the collected data at a single location creates a single point of failure and is associated with high security and privacy risks, motivating distributed data storage and processing. A distributed solution eliminates the single point of failure and allows for privacy-preserving data processing among multiple service providers via, for instance, secure multiparty computation (MPC). In this setting, the devices send encrypted data to multiple cloud service providers and employ a suitable secret sharing scheme to share the encryption key with them. The cloud servers then decrypt the encrypted data in a distributed manner to obtain shares of the plaintext data on which they can perform privacy-preserving analysis. One challenge in this context has to do with the resource-constrained nature of many IoT devices, necessitating the adoption of lightweight cryptographic mechanisms explicitly designed for such devices. However, the study of lightweight cryptographic mechanisms, such as lightweight ciphers, is lacking in the MPC context. So far, the de-facto benchmark for MPC has been the evaluation of the AES. Therefore, it is important to investigate the performance of lightweight ciphers in MPC, specifically, their implementation that accommodates the distributed decryption processes in the subsequent data processing. In this poster, we report on the work in progress to fill this gap by designing a new garbling scheme that allows very fast evaluation of non-linear functions such as S-boxes commonly found in substitution-permutation network ciphers (SPN). We show estimated evaluation performance for selected lightweight ciphers and report on the performance of a preliminary implementation of the SKINNY cipher family.

II. Preliminary Results

A. Garbling Scheme

We design a new garbling scheme as a primitive for two-party secure computation. Unlike traditional garbling schemes that compute on circuits with XOR and AND gates, and inspired from arithmetic garbling [1], we compute on circuits with wires representing n bits, XOR gates and unary projection gates that enable the computation of any n to m-bit function.

1) Wire label offsets: Let $\bar{n}$ denote the maximum bit-length of wires used in the circuit, then we use bit strings of length $k = \kappa + \bar{n}$ as wire labels where κ is the security parameter. With $\mathsf{lsb}_n(W)$ we denote the n least significant bits of the bitstring $W \in \{0,1\}^k$. For each bit-length n ($1 \leq n \leq \bar{n}$), a wire label offset is a bit-vector of length k with $k-n$ random bits and n fixed bits. The garbler draws n wire label offsets and groups those vectors by columns to form the matrix $\vec{R}_n \in \{0,1\}^{k \times n}$. We denote the column-vectors of $\vec{R}_n$ as

$$\vec{R}_n = \left(R_{0\ldots01}, R_{0\ldots010}, \ldots, R_{10\ldots0}\right),$$

where the fixed n bits are

$$\mathsf{lsb}_n(R_{0\ldots01}) = 0\ldots01, \ \ldots, \ \mathsf{lsb}_n(R_{10\ldots0}) = 10\ldots0 \ .$$

We use R_x and $x \cdot \vec{R}_n$ interchangeably when the choice of n is clear from context. The inner product of $x \cdot \vec{R}_n$ is defined as $x_0 R_{0\ldots01} \oplus \ldots \oplus x_{n-1} R_{10\ldots0}$.

2) Wire label encoding: The encoding W_i^x of an n-bit number $x \in \{0,1\}^n$ on a wire with index i is defined as

$$W_i^x = W_i^{0^n} \oplus x \cdot \vec{R}_n = W_i^{0^n} \oplus R_x \ .$$

Intuitively, there are n distinct offsets R, one for each encoded bit. The offset applied to a wire label that represents x is the linear combination of R values.

3) XOR gates: For an XOR gate with n-bit input wires a and b and output wire c, the garbler generates the output wire label

$$W_c^x \leftarrow \underbrace{W_a^{0^n} \oplus W_b^{0^n}}_{W_c^{0^n}} \oplus R_x \ ,$$

where $x \in \{0,1\}^n$. No ciphertext is sent.

Having obtained W_a and W_b as the input wire labels for the XOR gate, the evaluator computes $W_c \leftarrow W_a \oplus W_b$.

4) Projection gates: Let a be the input wire to the projection gate Proj_ϕ that computes the unary projection $\phi : \{0,1\}^n \mapsto \{0,1\}^m$, and c be the output wire. The garbler first draws the output wire label for 0 at random: $W_c^{0^m} \overset{\$}{\leftarrow} \{0,1\}^k$ and then generates 2^n ciphertexts for each $x \in \{0,1\}^n$:

$$GC[c, \mathsf{lsb}_n(W_a^x)] \leftarrow \mathcal{H}(W_a^x, c) \oplus W_c^{\phi(x)} \ .$$

All ciphertexts are sent to the evaluator[1]. Having obtained W_a as the input wire label to the projection gate, the evaluator computes the output label W_c by

$$W_c \leftarrow GC[c, \mathsf{lsb}_n(W_a)] \oplus \mathcal{H}(W_a, c) \ .$$

Assuming that the hash function $\mathcal{H}$ fulfils a generalization of circular correlation robustness, we obtain privacy and obliviousness in the BHR framework [3]. Authenticity can be added in a straightforward way.

B. Applications: SPN-based symmetric primitives

Here, we discuss the implementation of SPN-based symmetric primitives as one practical application. In these primitives, a state is split into n-bit cells and is updated with a round function consisting of a substitution layer, a permutation layer and a round constant and/or (round) key addition layer. We implement a single cell as n-bit wire. Each S-box in the substitution layer is replaced with an n-bit projection gate computing the same functionality. The permutation layer and the addition layer are expressible with XOR gates only.

Indeed, we identified many SPN primitives in literature that fulfil the conditions. They are listed in Table I with the corresponding trade-off in garbling and communication cost, and evaluation improvement measured in the number of calls to $\mathcal{H}$ and in the number of ciphertexts, respectively. We found several primitives where our scheme improves in both garbling and evaluation cost over both reference garbling schemes. In the remaining primitives and cases, projection gates trade-off higher garbling and communication cost for faster evaluation performance. Note that for most primitives, the evaluation improvement is much higher than the additional communication cost. For e.g. Midori64, we expect ≈ 2 to 2.6 times more ciphertexts that need to be sent but save a factor 5 to 8 in evaluation work, respectively.

C. Implementation

We evaluated an implementation of the SKINNY cipher family [2] with garbling schemes ZRE15 (named AND) and the projection gates scheme (named Proj) in MP-SPDZ [4] using Yao's semi-honest protocol implementation. In addition, we created an optimized implementation of parallel SKINNY encryptions based on AND. We denote this implementation Bit since it follows the style of bit-slicing. It must be noted that, still, the same number of gates, calls to the primitive etc., are performed, but due to internal memory layout and execution details of the MP-SPDZ compiler and virtual machines, this approach executes faster.

TABLE I. Estimated performance difference for selected symmetric primitives. The notation + x denotes an improvement by factor x in the respective category, the notation - x denotes a degradation by factor x.

Base Scheme	Primitive	Garble	Send	Eval
ZRE15	TWINE-128	+1.44	-1.28	+9.05
RR21		+2.16	-1.70	+13.58
ZRE15	Fides-96	+2.10	+1.07	+50.26
RR21		+3.15	-1.24	+75.39
ZRE15	WAGE	+1.51	-1.31	+72.87
RR21		+2.27	-1.75	+109.30
ZRE15	MANTIS	-1.11	-1.98	+4.31
RR21		+1.35	-2.64	+6.46
ZRE15	Midori64	-1.06	-1.94	+5.33
RR21		+1.41	-2.58	+8.00
ZRE15	CRAFT	-1.05	-1.93	+5.71
RR21		+1.43	-2.57	+8.57
ZRE15	SKINNY-64-128	-1.51	-2.81	+4.68
RR21		-1.01	-3.75	+7.02
ZRE15	Piccolo-128	-1.53	-2.84	+4.55
RR21		-1.02	-3.79	+6.83

TABLE II. Performance of circuit-based implementations of 100 SKINNY encryptions. Garble, Eval and Communication cost are amortized, i.e. given *per* block.

Cipher	Impl.	Garble (std)	Eval (std)	Comm.
		in ms	in ms	in KB
SKINNY-64-128	AND	0.70 (0.12)	0.56 (0.08)	74
	Bit	0.28 (0.05)	0.17 (0.04)	74
	Proj	0.35 (0.05)	0.07 (0.01)	139
SKINNY-128-128	AND	1.81 (0.23)	1.10 (0.15)	164
	Bit	0.78 (0.12)	0.32 (0.06)	164
	Proj	5.78 (0.33)	0.04 (0.01)	2613

The garble resp. eval column in Table II denote the average garbling resp. evaluation time of 200 runs for 100 blocks with the same key. The communication cost includes all ciphertexts that have to be sent from the garbler to the evaluator and data from oblivious transfer. Both garbler and evaluator executed on the same machine (6-core AMD Ryzen 5 PRO 2.1 GHz with 8 GB RAM) but were limited to use only 4 threads each. The projection gate implementation achieves the fastest evaluation time with a 2.4 to 25-fold speed-up. This comes at the price of an 1.25 to 7.4-fold increase in garbling time and an 1.8 to 15 times heavier communication cost.

Acknowledgment

This work is supported by the Flemish Government through FWO SBO project MOZAIK S003321N.

References

[1] M. Ball, T. Malkin, and M. Rosulek, "Garbling gadgets for boolean and arithmetic circuits," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 565–577.

[2] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 123–153.

[3] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 784–796.

[4] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[1] Here, garbled row reduction allows one less ciphertext to be sent.

Poster: Fast Evaluation of S-boxes in MPC

Erik Pohle, Aysajan Abidin, Bart Preneel

imec-COSIC, KU Leuven, Belgium

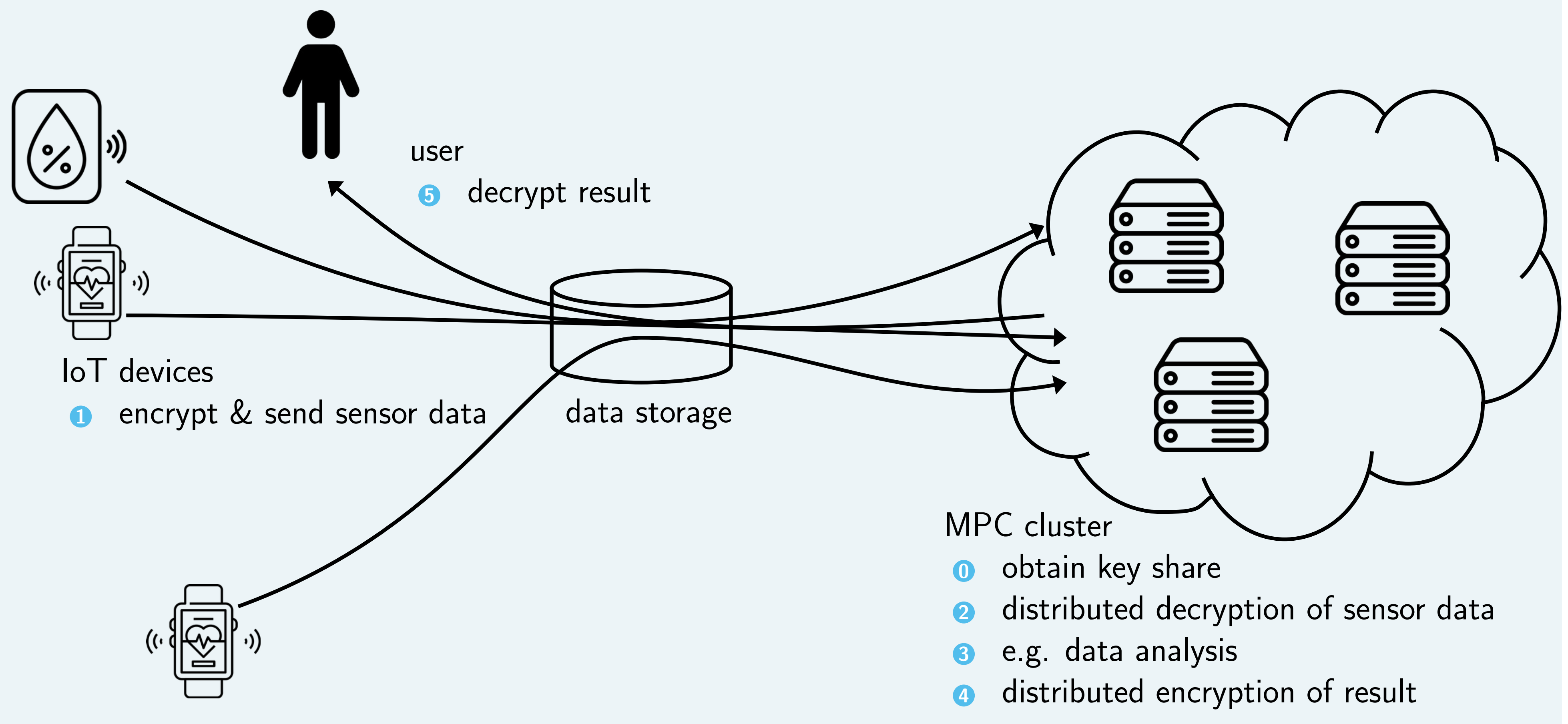
NDSS 2022

I Introduction

For secure IoT sensor data collection and processing, IoT devices send encrypted data to multiple cloud service providers and employ a suitable secret sharing scheme to share the encryption key with them. The cloud servers then decrypt the encrypted data in a distributed manner to obtain shares of the plaintext data on which they can perform privacy-preserving analysis. Due to the resource-constraint nature of IoT devices, it is important to investigate the performance of lightweight ciphers in secure multi-party computation (MPC), specifically, their implementation that accommodates the distributed decryption processes in the subsequent data processing.

In this poster, we report on the work in progress on a new garbling scheme that allows very fast evaluation of non-linear functions such as S-boxes commonly found in substitution-permutation network ciphers (SPN). We show estimated evaluation performance for selected lightweight ciphers and report on the performance of a preliminary implementation of the SKINNY cipher family.

II Use-Case



user

5 decrypt result

IoT devices

1 encrypt & send sensor data

data storage

MPC cluster

0 obtain key share

2 distributed decryption of sensor data

3 e.g. data analysis

4 distributed encryption of result

Icons made by Pixel perfect, LAFS, BZZRINCANTATION and Freepik from www.flaticon.com

III A new garbling scheme

We design a new garbling scheme as a primitive for two-party secure computation. Unlike traditional garbling schemes that compute on circuits with XOR and AND gates, and inspired from arithmetic garbling [1], we compute on circuits with wires representing n bits, XOR gates and unary projection gates that enable the computation of any n to m-bit function.

Wire Labels

The encoding W_i^x of an n-bit number $x \in \{0,1\}^n$ on a wire with index i is

$$W_i^x = W_i^{0^n} \oplus x \cdot \vec{R}_n = W_i^{0^n} \oplus R_x \ .$$

Intuitively, there are n distinct offsets R, one for each encoded bit. The offset applied to a wire label that represents x is the linear combination of R values. We denote the column-vectors of $\vec{R}_n$ as

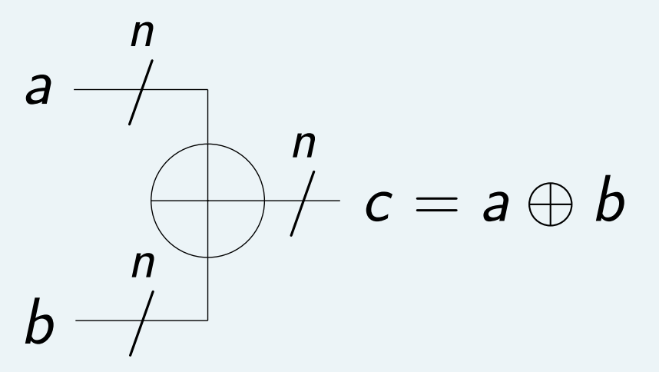
$$\vec{R}_n = (R_{0\ldots01}, R_{0\ldots010}, \ldots, R_{10\ldots0}) \, ,$$

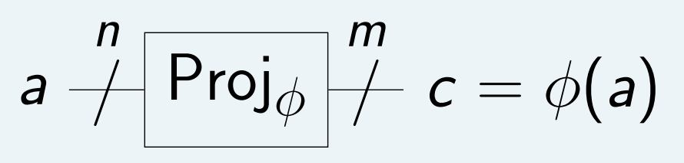
where the fixed n bits are

$$\mathrm{lsb}_n(R_{0\ldots01}) = 0\ldots01, \ \ \ldots, \ \mathrm{lsb}_n(R_{10\ldots0}) = 10\ldots0 \ .$$

The remaining $k - n$ bits are randomly chosen by the garbler.

Gates





For an XOR gate with n-bit input wires a and b and output wire c, the garbler generates the output wire label

$$W_c^x \leftarrow \underbrace{W_a^{0^n} \oplus W_b^{0^n}}_{W_c^{0^n}} \oplus R_x \ ,$$

where $x \in \{0,1\}^n$. No ciphertext is sent.

Having obtained W_a and W_b as the input wire labels for the XOR gate, the evaluator computes $W_c \leftarrow W_a \oplus W_b$.

The projection gate Proj_ϕ computes the unary projection $\phi : \{0,1\}^n \mapsto \{0,1\}^m$ on the n-bit input wire a with output wire c.

1. The garbler draws the output wire label for 0 at random: $W_c^{0^m} \overset{\$}{\leftarrow} \{0,1\}^k$.
2. The garbler generates 2^n ciphertexts for each $x \in \{0,1\}^n$:

$$GC[c, \mathrm{lsb}_n(W_a^x)] \leftarrow \mathcal{H}(W_a^x, c) \oplus W_c^{\phi(x)} \ .$$

All ciphertexts are sent to the evaluator. Having obtained W_a as the input wire label to the projection gate, the evaluator computes the output label W_c by

$$W_c \leftarrow GC[c, \mathrm{lsb}_n(W_a)] \oplus \mathcal{H}(W_a, c) \ .$$

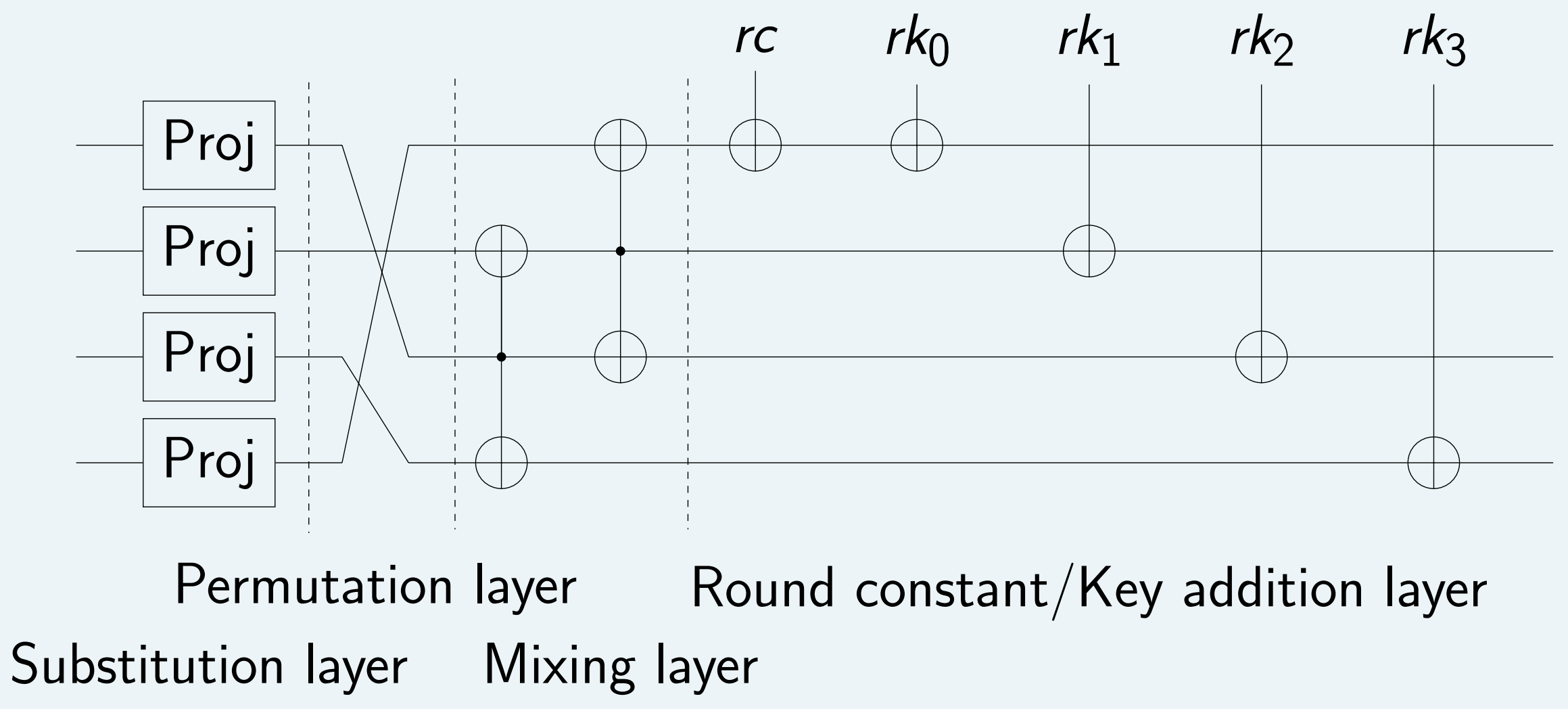
Security

Assuming that the hash function $\mathcal{H}$ fulfils a generalization of circular correlation robustness, we obtain privacy and obliviousness in the BHR framework [3]. Authenticity can be added in a straight-forward way.

IV Application: SPN-ciphers

The following conditions for state and round function parts are sufficient to ensure a well-performing circuit representation with projection gates.

- **State.** The state is (conceptionally) split into n-bit cells.
- **Substitution Layer.** The substitution layer consists of S-boxes that are applied to each cell.
- **Permutation Layer.** The permutation layer can be described by a permutation on the cells and/or by a binary mixing matrix which encodes a fixed matrix multiplication with the state.
- **Round constant/(round) key addition layer.** The round constant or (round) key is XORed cell-wise.



Permutation layer

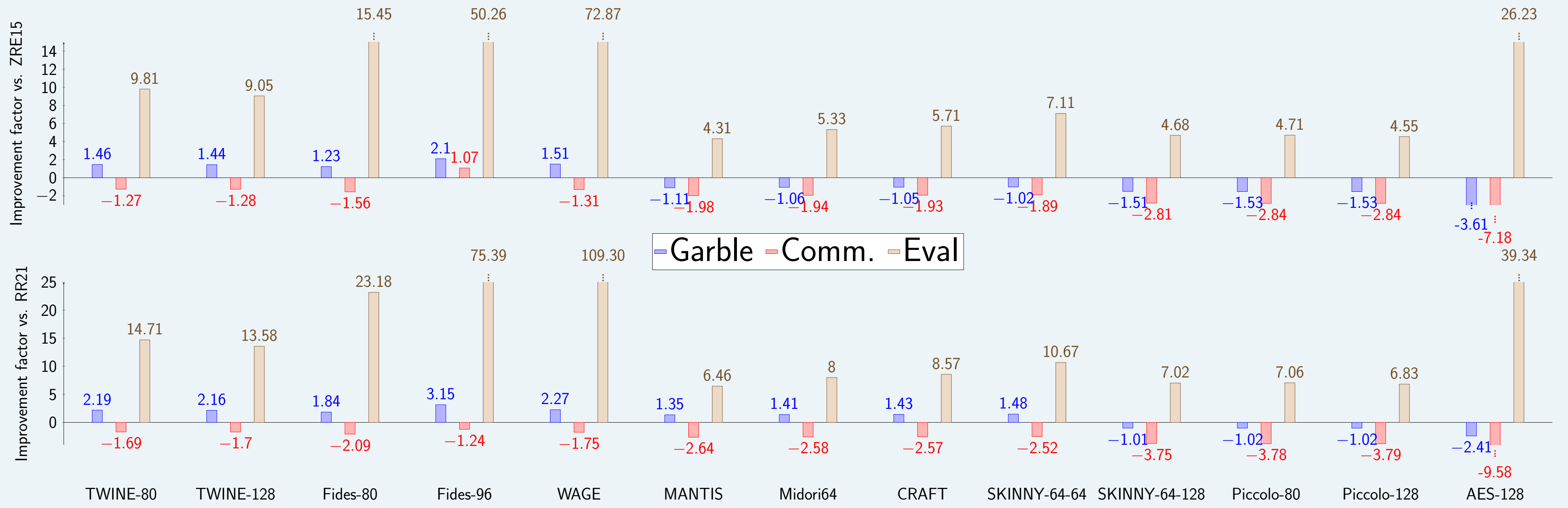
Round constant/Key addition layer

Substitution layer

Mixing layer

V Results

Estimated performance difference for selected symmetric primitives.



VI Implementation

Evaluation of the SKINNY cipher family [2] in MP-SPDZ [4] using Yao's semi-honest protocol implementation.

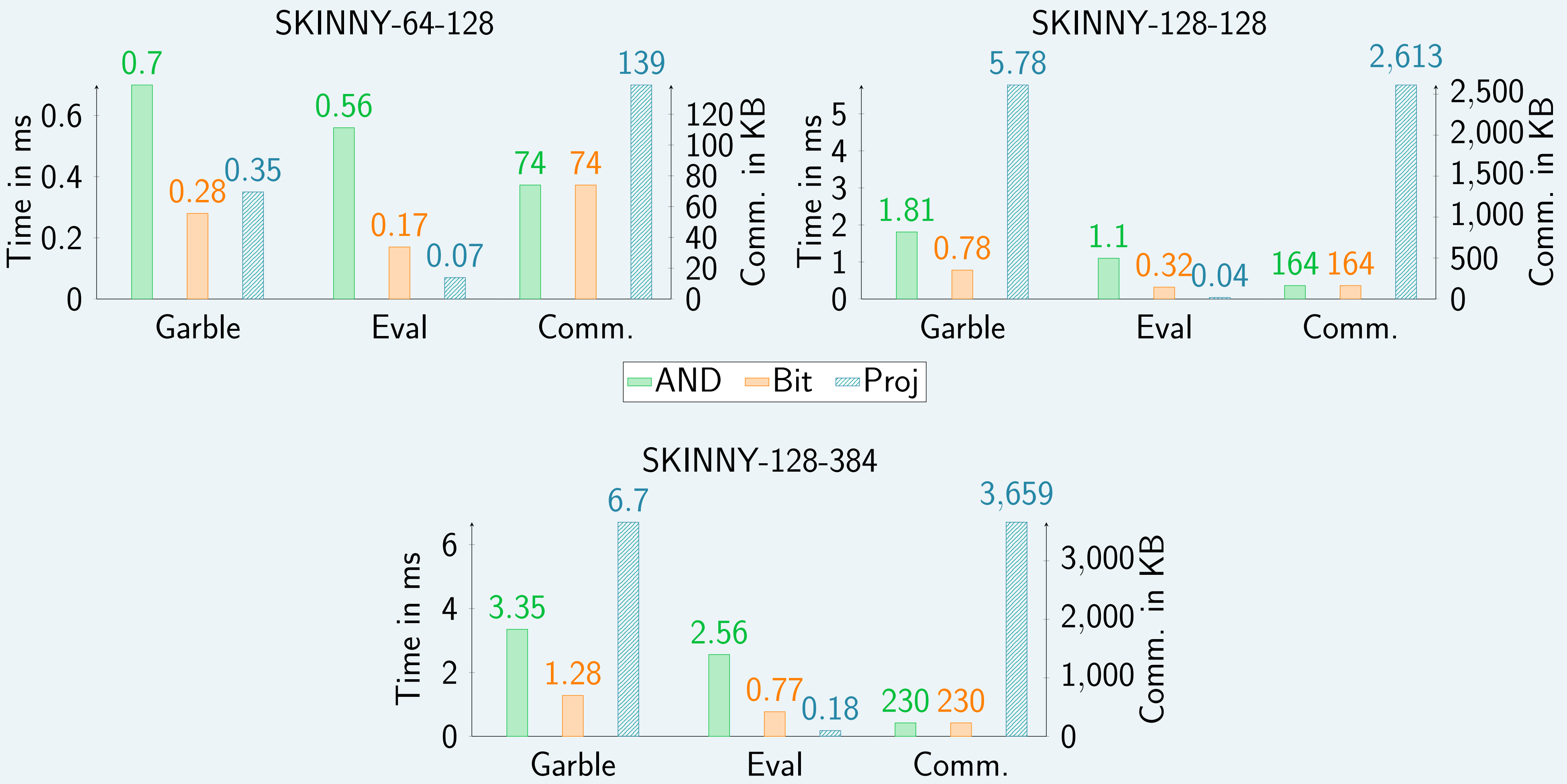
Implementation Variants

AND (▭) Garbling scheme ZRE15, i.e. HalfGates and FreeXOR

Bit (▭) Optimized implementation of parallel SKINNY encryptions based on AND. Speed-up is due to efficient usage of internal memory layout and execution details of the MP-SPDZ compiler and virtual machines.

Proj (▨) The new projection gates scheme.

Benchmark Results



- Performance of 100 SKINNY encryptions, all costs are amortized, i.e. given *per* block, averaged over 200 runs. The communication cost includes all ciphertexts and data from oblivious transfer
- Benchmark on the same machine (6-core AMD Ryzen 5 PRO 2.1 GHz with 8 GB RAM) limited to 4 threads per party

→ Variant Proj (▨) achieves the fastest evaluation time with a 2.4 to 25-fold speed-up at the price of an 1.25 to 7.4-fold increase in garbling time and an 1.8 to 15 times heavier communication cost.

References

[1] M. Ball, T. Malkin, and M. Rosulek, "Garbling gadgets for boolean and arithmetic circuits", in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 565–577.

[2] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, "The SKINNY family of block ciphers and its low-latency variant MANTIS", in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 123–153.

[3] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits", in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 784–796.

[4] C. M. Keller, "MP-SPDZ: A versatile framework for multi-party computation", in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

This work is supported by the Flemish Government through FWO SBO project MOZAIK S003321N.