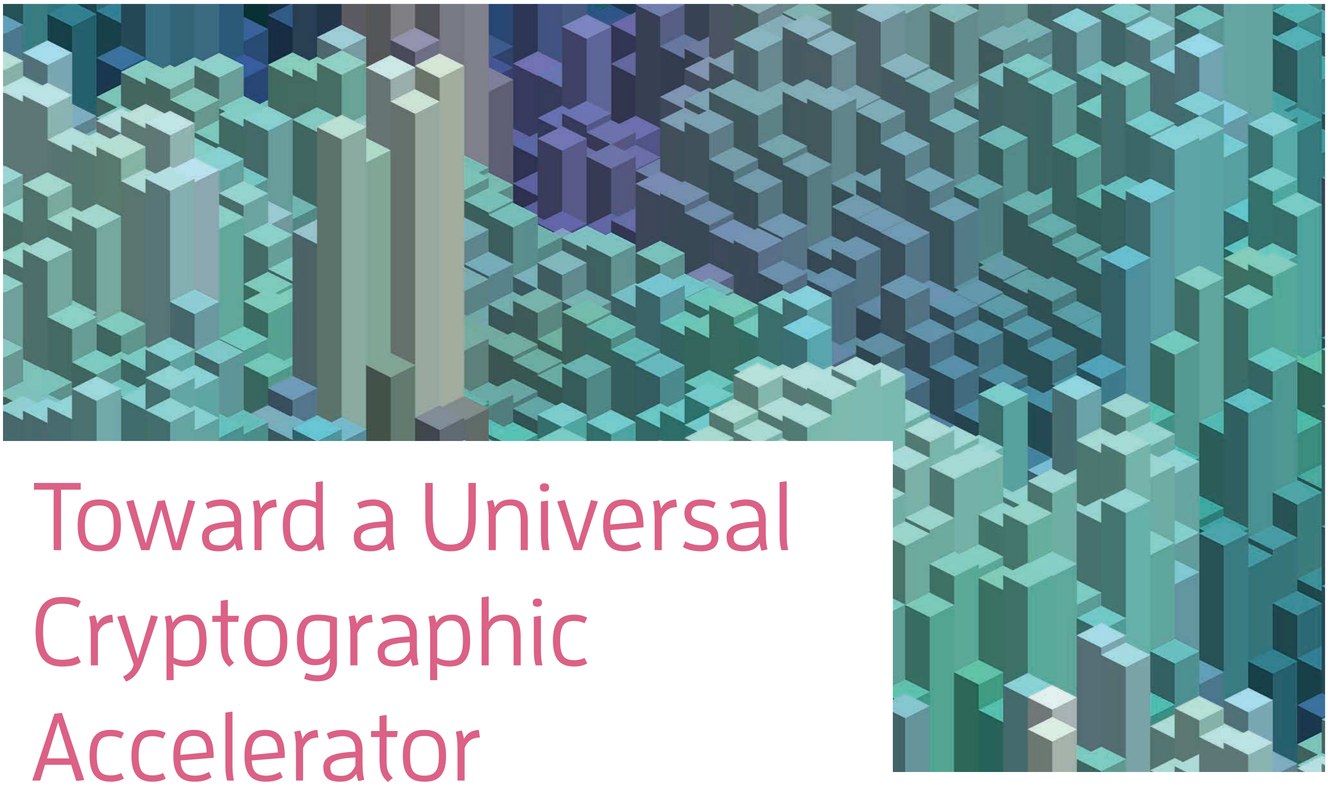
CYBER-PHYSICAL SYSTEMS

EDITOR DIMITRIOS SERPANOS
CTI DIOPHANTUS and University of Patras;
serpanos@computer.org



Toward a Universal Cryptographic Accelerator

Srinivas Devadas and Daniel Sanchez, Massachusetts Institute of Technology

Cyberphysical systems have disseminated devices that can be untrustworthy or compromised. Nevertheless, the privacy and integrity of computation and data can be guaranteed through cryptographic protocols. We address the computational burden posed by cryptography, and argue for a synergistic approach of designing programmable hardware accelerators for cryptography, followed by tailoring cryptographic protocols to this hardware.

Cryptographic technologies for data encryption and authentication are mature and pervasive on the Internet [for example, Transport Layer Security (TLS), Secure Sockets Layer (SSL), and HTTPS]. But these cryptographic techniques protect data only during transmission and storage, not processing. Cyberphysical systems feature diverse and disseminated devices that process sensitive data and perform critical functions—but whose hardware or software can be easily compromised by an attacker. We propose to secure these systems by taking an approach that is rooted in cryptography and does not require trust in the hardware.

Sophisticated cryptographic primitives, such as fully homomorphic encryption (FHE), zero-knowledge proofs (ZKPs), and private information retrieval (PIR), are increasingly

Digital Object Identifier 10.1109/MC.2024.3481808
Date of current version: 9 January 2025

being used in varied applications such as outsourced computation, cryptocurrencies, private search, and anonymous communication. These primitives provide privacy and integrity guarantees on data processing, enabling secure and reliable computation from untrusted devices or servers. Unfortunately, state-of-the-art implementations of these primitives suffer poor performance, which limits their applicability: FHE and ZKP incur slowdowns ranging from four to six orders of magnitude over native computation, limiting them to small programs. Similarly, PIR schemes require time proportional to the size of the database in comparison to logarithmic lookup in a nonprivate setting.

We therefore espouse a synergistic approach of hardware designed for cryptography, which in turn is designed with hardware in mind.

Our goal is to take a significant step toward universal cryptographic acceleration infrastructure. We are developing a hardware architecture and a compiler that speed up a broad range of cryptographic protocols by multiple orders of magnitude. We are aided by the recent rise of postquantum-secure lattice cryptography, which is producing best-in-class protocols over a wide range of cryptographic applications, including those discussed previously. If successful, these protocols will become widely applicable and further jumpstart the development of new cryptographic protocols that exploit and depend on hardware acceleration.

Our approach seeks to achieve the same synergy and wide impact for emerging cryptography that Advanced Encryption Standard (AES) hardware acceleration has had for established cryptographic primitives. While AES is a symmetric-key encryption standard,[1] since the introduction of the AES New Instructions (AES-NI) hardware instruction set in the early 2010s, there has been tremendous growth in cryptographic software taking advantage of hardware-accelerated AES for primitives other than encryption, including keyed hash functions and pseudorandom number generators.

We therefore espouse a synergistic approach of hardware designed for cryptography, which in turn is designed with hardware in mind. As one concrete effort, consider the complementary goals of FHE and ZKP. FHE guarantees the privacy of client data throughout the computation, and ZKP guarantees the integrity of the computation (while protecting the privacy of particular inputs as needed). There are many schemes for FHE, some more amenable to acceleration than others.

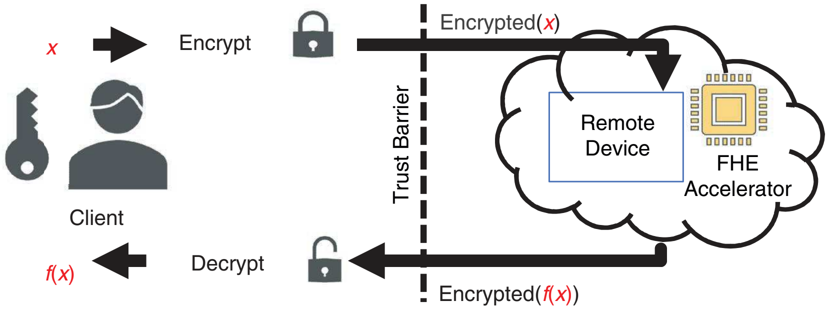


FIGURE 1. FHE can offload computation to an untrusted device securely.

The same is true for ZKP. Can ZKP schemes be tailored to fit a hardware accelerator for FHE? We argue in this article that the answer is yes.

BACKGROUND

Figure 1 shows how FHE enables the secure offloading of computation. The client wants to compute a function *f*(·) (for example, a deep learning inference) on some private data *x*. Computing *f*(·) locally may be too expensive for the client (for example, each inference requires either too many operations or a too-large model). To do this, the client encrypts *x* and sends it to an untrusted device, which computes *f*(·) on these encrypted data directly using FHE and returns the encrypted result to the client. FHE provides ideal privacy; even if the device is compromised, attackers cannot learn anything about the data *x* as they remain encrypted throughout.

ZKPs are an emerging family of cryptographic tools enabling one party (the prover) to prove to other parties (the verifiers) that a statement is true without requiring the prover to disclose any data to the verifiers. Figure 2 illustrates how ZKPs work. A prover generates a small proof of a statement and publishes it. Any verifier can download the proof and verify the statement cheaply. The proof is zero knowledge in that the private witness is not exposed to the verifier; however, this witness is an *optional* feature. One use case of ZKPs is complementary to that of FHE; a client wants to offload the computation of *f*(*x*) to a remote device. The device returns the output *y* and a proof of correctness that the client can use to verify that *f*(*x*) was computed correctly. Verifying the proof is orders of magnitude cheaper than computing the proof—and usually cheaper than computing *f*(*x*) locally. Unfortunately, proof generation, which happens in the remote device, is currently orders of magnitude more expensive than computing *f*(*x*); this bottlenecks ZKP schemes and is a prime target for hardware acceleration (Figure 2). In summary, on untrusted devices, ZKP provides integrity

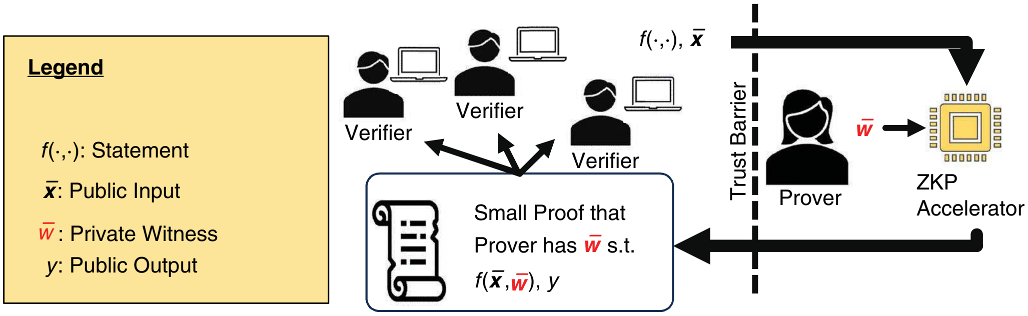

Legend
$f(\cdot,\cdot)$: Statement
$\bar{x}$: Public Input
$\bar{w}$: Private Witness
y: Public Output
Verifier
Verifier
Verifier
$f(\cdot,\cdot)$, $\bar{x}$
Trust Barrier
Prover
$\bar{w}$
ZKP Accelerator
Small Proof that
Prover has $\bar{w}$ s.t.
$f(\bar{x},\bar{w})$, y

FIGURE 2. ZKPs allow a prover to convince other parties, called *verifiers*, of a statement *f*, optionally requiring private input *w* from the prover.

of computation, whereas FHE provides privacy without integrity guarantees.

HARDWARE ACCELERATION

Our team designed the F1[2] and CraterLake[3] FHE accelerators in 2021 and 2022, respectively. F1 speeds up shallow FHE programs, and CraterLake accelerates unbounded-depth FHE programs by 5,000× over a 32-core CPU and scales to large chips efficiently. We contributed several techniques that make this possible, including

1. a new extremely wide (2,048 lanes) vector uniprocessor architecture that spreads each vector operation across the chip, departing from prior vector multicore architectures
2. an efficient implementation of this architecture, for non-single optimized instruction, multiple data FHE operations, number-theoretic transforms (NTTs), and automorphisms by effectively decomposing them among distributed groups of lanes using a novel transpose network.

CraterLake is a current state-of-the-art FHE accelerator, and its features have been used in subsequent designs.

Suppose one wished to build a unified hardware accelerator that handles the full range of emerging cryptography, including FHE, ZKP, and PIR. Accelerators are expensive to design and build, so chips that can be used on many applications have a much higher probability of impact and wide adoption. At first glance, it seems challenging to build a single accelerator for this broad range of cryptographic protocols. Our insight is that, while diverse, many of these protocols share very similar computational characteristics; all work on large operands, require high-throughput modular arithmetic, and perform a similar set of primitive operations (like NTTs). These shared features, which make these protocols slow on CPUs and GPUs, can also enable a single accelerator to handle them.

CraterLake is a current state-of-the-art FHE accelerator, and its features have been used in subsequent designs.

As a first step toward a unified accelerator, we have designed a novel accelerator, NoCap,[4] that leverages hardware/algorithm co-design to achieve transformative speedups. NoCap generates proofs 586× faster than a 32-core CPU and 41× faster than PipeZK, a state-of-the-art ZKP accelerator. We leverage recent algorithmic developments to achieve these speedups; we identify and combine two recent hash-based ZKP algorithms, Orion and Spartan, which have similar performance on CPUs to the ZKPs targeted by prior accelerators, but are much more amenable to hardware acceleration. We chose Orion and Spartan rather than more popular elliptic-curve-based schemes (for example, Groth16) because the operations in these schemes have greater similarity to those in FHE schemes. Though Orion and Spartan result in larger proofs, we have shown that, for many applications, the end-to-end speedups (including prover time, proof transmission, and verification time) more than justify this size increase. We developed a novel hardware organization to exploit these acceleration opportunities; NoCap is a programmable vector processor with functional units tailored to the needs of hash-based ZKPs. We then combined Spartan and Orion as a novel way to form what we call the *Spartan+Orion ZKP*, which is an excellent fit for our accelerator; additional optimizations improve parallelism and reduce memory traffic. As a result, NoCap achieves speedups that enable new use cases for ZKP.

NoCap shares similarities with FHE accelerators because hash-based ZKPs and FHE schemes both consist of regular computations on large polynomials with modular integer coefficients. This results in similar functional units, for example, to perform element-wise arithmetic operations and NTTs, and paves the way for future chips that efficiently accelerate FHE, ZKPs, and other cryptographic protocols that rely on lattice-based cryptography.

We believe a successful marriage of hardware acceleration and cryptography has a high potential for impact; by lowering the cost of cryptographic techniques, these techniques will become an essential part of the secure cyberphysical systems of the future, saving billions of dollars and enabling security in a sustainable, energy-efficient way.

REFERENCES

1. J. Nechvatal et al., “Report on the development of the Advanced Encryption Standard (AES),” *J. Res. Nat. Inst. Standards Technol.*, vol. 106, no. 3, p. 511, Jun. 2001, doi: 10.6028/jres.106.023.
2. A. Feldmann et al., “F1: A fast and programmable accelerator for fully homomorphic encryption,” in *Proc. 54th Int. Symp. Microarchit. (MICRO)*, Virtual Event, Greece, Oct. 2021, pp. 238–252, doi: 10.1145/3466752.3480070.
3. N. Samardzic et al., “CraterLake: A hardware accelerator for efficient unbounded computation on encrypted data,” in *Proc. 49th Int. Symp. Comput. Archit. (ISCA)*, New York, New York, Jun. 2022, pp. 173–187.
4. N. Samardzic, S. Langowski, S. Devadas, and D. Sanchez, “Accelerating zero-knowledge proofs through hardware-algorithm co-design,” in *Proc. 57th Int. Symp. Microarchit. (MICRO)*, Austin, Texas, USA, Nov. 2024.

SRINIVAS DEVADAS is the Webster Professor of Electrical Engineering and Computer Science at MIT, Cambridge, MA 02139 USA, where he has been on the faculty since 1988. Contact him at devadas@mit.edu.

DANIEL SANCHEZ is a professor of electrical engineering and computer science at MIT, Cambridge, MA 02139 USA, where he has been on the faculty since 2012. Contact him at sanchez@csail.mit.edu.

Digital Object Identifier 10.1109/MC.2024.3516459