Off-chaining Models and Approaches to Off-chain Computations

Jacob Eberhardt
Information Systems Engineering, TU Berlin
Germany
je@ise.tu-berlin.de

Jonathan Heiss
Information Systems Engineering, TU Berlin
Germany
jh@ise.tu-berlin.de

ABSTRACT

Off-chaining has been suggested to enhance scalablity and privacy of blockchains, especially in public networks. However, a systematic classification is missing. In this paper, we propose generic off-chaining models categorizing different types of off-chaining. From our analysis of these models, we conclude that off-chain computations are particularly powerful and subsequently provide a description and comparison of off-chain computation approaches.

KEYWORDS

blockchains, scalability, privacy, off-chain, zkSNARKs, zkSTARKs, Bulletproofs, TEE, sMPC, Incentives

ACM Reference Format:
Jacob Eberhardt and Jonathan Heiss. 2018. Off-chaining Models and Approaches to Off-chain Computations. In *2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers (SERIAL'18), December 10–14, 2018, Rennes, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3284764.3284766

1 INTRODUCTION

Blockchains represent a novel class of distributed systems with unique properties and capabilities. They enable the execution of programs without requiring trust in a single party, store an immutable history of transactions, and provide a highly available and censorship resistant platform for decentralized applications. However, besides legal, cultural, and organizational challenges, especially technical limitations currently prevent web-scale blockchain-based systems from becoming reality.

Every transaction is processed at every node in the network, which limits throughput. The performance impact is especially severe in public blockchain networks. Furthermore, to allow independent transaction validation, all data needs to be available at every node. Thus, privacy is not guaranteed in public blockchain networks like Bitcoin [24] or Ethereum [10].

Off-chaining has been suggested [12] as one approach to address these limitations. The idea is to reduce computational efforts and data storage on the blockchain by employing blockchain-external resources. Yet, key properties introduced by blockchain technology, e.g., immutability and availability, must not be compromised.

SERIAL'18, December 10–14, 2018, Rennes, France

ACM ISBN 978-1-4503-6110-1/18/12…$15.00
https://doi.org/10.1145/3284764.3284766

To date, it is hard to compare and reason about different off-chaining approaches as no systematic classification exists. Hence, as our first contribution, we introduce conceptual off-chaining models which help to categorize different off-chaining approaches. We distinguish between Off-chain Storage, Off-chain Computation and Hybrid Off-chaining.

Storing data off the blockchain could help to reduce storage consumption for clients and enhance confidentiality by hiding data from other nodes in the network. However, data availability cannot be guaranteed and hence data off-chaining requires careful consideration in order to not impair liveness. Off-chain computations could help to reduce processing redundancy and with that scalability, but need to leverage mechanisms that do not reintroduce trust as an assumption.

Based on our analysis, we conclude that off-chain computations are a particularly promising research area to address privacy and scalability of blockchain networks. Blockchains should not compute, but provide data in a highly available way, while state transitions are computed off-chain.

Due to the novelty and complexity of approaches suitable for off-chain computations, it is hard to assess their properties, compare them, and assess their applicability for concrete use cases. Thus, as our second contribution, we provide an overview of different off-chain computation approaches that are currently discussed and compare them with regards to scalability, privacy, security and programmability. For that purpose, we classify approaches into the categories Verifiable Computation, Secure Multiparty Computation, Enclave-based Computation, and Incentive-driven Computation.

To the best of our knowledge, there is no related work either deriving a classification for off-chaining approaches or providing a comprehensive comparison of off-chain computation proposals.

2 OFF-CHAINING MODELS

In this section, we introduce a generic off-chaining model which is derived from on-chain transaction processing and describes off-chain storage, computation, and hybrid approaches combining the two.

2.1 On-chain Transaction Execution

Blockchains can be envisioned as global state machines that process transactions leading to state transitions [32]. A chosen consensus protocol establishes a globally unique order of transactions which are then executed redundantly on every node in a blockchain network.

Looking at the execution of a blockchain transaction more closely, it can conceptually be split into two distinct phases as depicted in Figure 1.

(i) Based on current state and inputs provided as part of the transaction, the *Compute* phase calculates a new state.
(ii) After the new state was computed, the *Persist* phase applies this state, i.e., writes it to the blockchain's persistent storage.

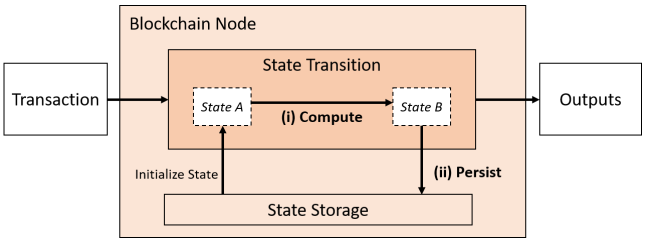

Figure 1: Blockchain Transaction Processing

While this model is quite simple, it emphasizes a key design decision in current blockchain systems: Computing new state and persisting it in the blockchain are tightly linked.

For every transaction processed by the blockchain, both operations are redundantly executed on every node in the network. The Compute step puts a computational burden on these nodes, whereas the Persist step mainly consumes storage space.

Hence, to reduce the cost for individual nodes and thus for the overall system, it could be beneficial to execute a phase in a less redundant way, i.e., off the blockchain. We refer to this idea as off-chaining. For this to be beneficial, it is crucial not to compromise the blockchain's key properties, e.g., immutability and availability.

2.2 Off-chain Storage

We define off-chaining storage as storing data on an Off-chain Node as depicted in Figure 2. An *Off-chain Node* is an arbitrary node not necessarily part of the blockchain network.

Before an on-chain state transition is computed, data is received by the blockchain from the storage node. This step relies on an external party writing the data to the blockchain. By themselves, blockchains cannot issue external calls since that would introduce non-determinism. On retrieval, data integrity is ensured in the Verify step. Finally, the Persist phase triggers an external party to write the new state back to the storage node.

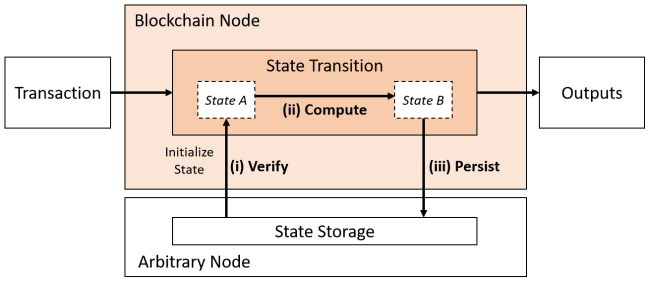

Figure 2: Transaction Processing with Off-Chain Storage

2.3 Off-chain Computation

We define off-chain computation as the execution model where the state transition function is computed by an Off-chain Node and the resulting state then persisted on-chain after verification of the computation of the state transition.

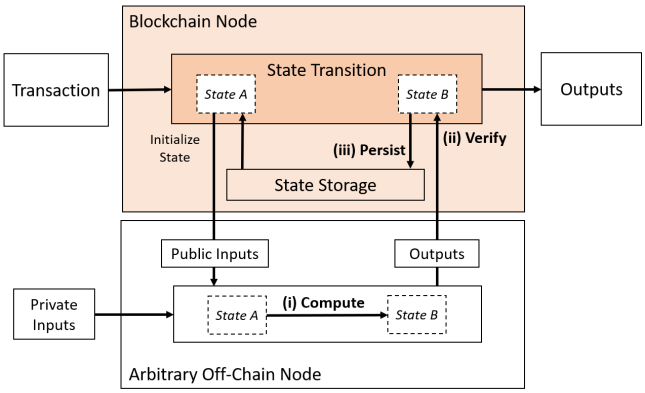

Figure 3: Transaction Processing with Off-Chain Computations

As depicted in Figure 3, in the first step an arbitrary off-chain node retrieves the relevant state from the blockchain that is required to compute the state transition. Since this information is stored on the blockchain, it is public and does not need to remain private during computation. Such inputs are referred to as *Public Inputs*. In contrast, the off-chain computation could also involve information that should remain private during and after execution. Such information can be provided as *Private Inputs*. Based on this input information, the off-chain node computes the state transition which results in *Outputs* that are then sent to the blockchain. The blockchain verifies the results and - if the verification succeeds - persists the new state to the on-chain state storage.

Off-chaining computations naively introduces a trust problem: What if the node executing the state transition lies about the result? To prevent this, the Verify phase is of utmost importance. A suitable off-chain computation mechanism needs to allow blockchain-based verification of the results.

2.4 Hybrid Off-chaining

We define the hybrid off-chaining model as the set of designs that combine off-chain state and off-chain computations in arbitrary ways and potentially in conjunction with on-chain processing. Figure 4 provides a high level overview of such a model and its components.

While it is hard to make generic statements regarding the properties of such hybrid approaches due to the vast number of possible designs, we briefly describe two relevant proposals as examples for this category.

State Channels, as implemented by the Lightning Network [27] or Raiden [21], combine off-chain state storage and off-chain state transitions. State transition logic is not necessarily codified but implicitly agreed on by participants and confirmed by their signatures. In the Verify Phase, these signatures are checked and in case

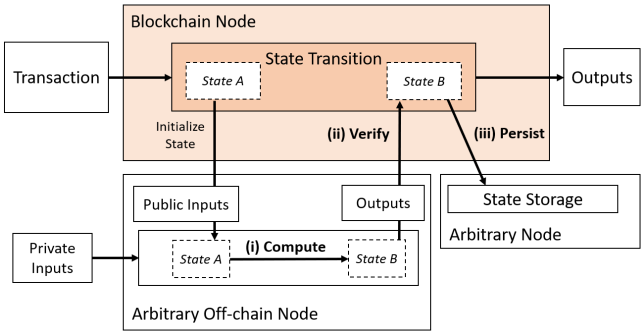

Figure 4: Components of a hybrid off-chaining model

of success the state is updated to the last version provided. This way, the blockchain ensures that invariants relevant to the whole network - for example that the sum of balances is equal before and after off-chain state transitions occurred - are preserved without validating state transitions themselves.

Plasma [26] combines off-chain state storage and off-chain computations by building a hierarchy of blockchains to achieve high scalability. Child blockchains compute state transitions and persist state independently but anchor it in parent chains periodically. Based on this, state transitions can be enforced in the parent blockchain, hence guaranteeing child chains' correctness.

A further exploration of this potentially large and impactful, yet rather unexplored set of off-chaining approaches is left for future work.

3 VIABILITY AND GENERALITY

In this section, we analyze the properties of the off-chain storage and computation models and conclude that off-chain computations are particularly promising when it comes to addressing the scalability and privacy challenges blockchain systems are facing today.

Off-chaining data from the blockchain involves two main challenges, data integrity and availability.

Data integrity needs to be preserved at all times, i.e., the blockchain needs to be able to verify on reception that the data has not been altered while stored off-chain or during transmission. This challenge can be solved elegantly by addressing off-chain data objects by their content - usually by the hash of data and storing such a hash pointer on the blockchain. By doing so, the blockchain can recalculate a data object's address by applying a hash function to the data received and comparing it to the on-chain reference.

Data availability is a harder problem to solve: Blockchains are highly available since all network nodes store the full blockchain state. Hence, every up-to-date node has guaranteed access to relevant state it requires for transaction validation. With off-chain state, this is no longer the case where blockchain transactions rely on data provided by an external node. Using a decentralized storage system with redundancy, e.g., IPFS [5] or SWARM [31], availability can be improved, but the blockchain's liveness guarantee is still weakened. To further enhance availability of off-chain data, approaches based on economic incentives, e.g., filecoin [22], have been proposed, but data loss or unavailability due to network partitions can still occur.

From this analysis, we infer that off-chain data storage can be useful in certain situations, but - unless the data availability problem is solved - cannot serve as a general purpose solution to reduce storage space of blockchain nodes. Off-chaining data is especially beneficial in scenarios where the blockchain is used to provide a reference to an off-chain data set, but without the need to be aware of the data itself. In contrast, it is certainly undesirable to prevent Smart Contracts [10] from making progress because of unavailable off-chain state information that needs to be retrieved and verified before the Compute phase.

Unlike with off-chain storage, liveness can still be guaranteed with off-chain computations: As long as off-chain state transitions are designed to be independently computable, nodes can always create transactions and do not need to rely on any third party.

In summary, we conclude that blockchains should not impair availability by off-chaining state and hence risk liveness, but should rather act as highly available storage that verifies state transitions computed elsewhere.

In the remaining sections of this paper, we present and discuss approaches that allow off-chaining computations without strong trust assumptions and benefit scalability as well as privacy to various degrees. Such approaches help to design more scalable and privacy-preserving blockchain systems. If verification of the off-chain computation results is cheaper than native on-chain execution, transaction throughput can be increased. Furthermore, private information can still influence the computation of state transitions by using Private Inputs and hence increase privacy and confidentiality.

4 OFF-CHAIN COMPUTATION APPROACHES

In this section, we introduce different approaches to off-chain computations and describe concrete implementations.

4.1 Verifiable Off-chain Computation

As the first computation off-chaining approach, we present Verifiable Off-chain Computation.

4.1.1 *Concept.*

We define Verifiable Off-chain Computation as an off-chaining technique where a *Prover* executes a computation and then publishes the result including a cryptographic proof attesting the computation's correctness to the blockchain. An on-chain *Verifier* then verifies the proof and persists the result in case of success.

There are many variants of verifiable computation schemes; we focus on the schemes particularly suitable to off-chaining. We identified these based on the following requirements, which we consider to be especially relevant in that context:

Non-interactivity: A Prover should be able to convince a Verifier in one message. Interactive schemes requiring multiple messages imply multiple blockchain transactions which increases load on the blockchain network and increases verification cost.

Cheap Verification: On-chain verification should be cheap compared to native on-chain execution. Otherwise, there would be no scalability benefit. If confidentiality, however, was the motivation for off-chaining a computation, additional cost over on-chain execution can be acceptable. The two factors influencing the cost

of on-chain verification are proof size and verification complexity. Ideally, verification cost is independent of the computational complexity of the off-chain computation.

Weak Security Assumptions: Security Assumptions should be as weak as possible. Strong assumptions would require additional trust which is in conflict with the blockchain paradigm.

Zero-knowledge: Schemes with the zero-knowledge properties allow Private Inputs to a computation, that never become public. While desirable for privacy and confidentiality, this property is not crucial to realize scalability benefits through off-chaining.

4.1.2 Realizations.

zkSNARKs: ZkSNARKs [15] allow a Prover to convince a Verifier that it executed a program correctly in one message while guaranteeing zero knowledge. The verification is cheap and its complexity is independent of the complexity of the computation to be proven.

Computations are specified as arithmetic circuits [25] or Rank-1-Constraint-Systems (R1CS) [4], which are hard to specify. However, higher level languages that compile into these abstractions have been developed [13, 20].

ZkSNARKs require a one-time setup step to be performed for a given program by a trusted party before executing off-chain computations. A malicious trusted party could create fake proofs. To weaken this trust assumption, multiparty computation protocols have been proposed [3, 6, 7] which distribute the setup phase over a set of nodes and are secure as long as there is at least one honest participant.

Bulletproofs: Bulletproofs [9] is a non-interactive zero-knowledge verifiable computation scheme that does not require a trusted setup. They were initially designed to enable efficient range proofs for confidential transactions, but can provide proofs for generic arithmetic circuits.

zkSTARKs: Like Bulletproofs, zkSTARKs [2] are a non-interactive zero-knowledge proof system that does not require a trusted setup. They do not rely on the arithmetic circuit abstraction but reduce to a set of higher degree polynomials. Because of that, there is no tooling available to conveniently specify provable programs. Currently, due to the immense prove size, practicality in blockchain applications is limited.

4.2 Enclave-based Off-chain Computation

In this section, we introduce Enclave-based Off-chain Computation (EOC) which relies on Trusted Execution Environments (TEE) to execute computations off-chain.

4.2.1 Concept.

TEEs are envisioned to guarantee confidential and integral code execution. A TEE is implemented as a secure part of the processor that protects a dedicated address space from unauthorized access. To guarantee the enclave's authenticity, an attestations certified by a trusted external entity is attached to every message.

Off-chain computations in EOC systems are exclusively executed inside the trusted enclave. To initiate a computation, Public Inputs are obtained from the blockchain and Private Inputs are optionally added by the Off-chain Node. Integrity of the Outputs is validated on-chain by verifying the enclave's attestation. Once verified, the new state is persisted to the blockchain's State Storage. To ensure confidentiality, state can be encrypted.

4.2.2 Realizations.

Enigma [29] and Ekiden [11] present two different implementations of EOCs. In Enigma programs can either be executed on-chain or in enclaves that are distributed accross a separate off-chain network. An Engima-specific scripting language allows developers to mark objects as private an hence, enforce off-chain computation. In contrast to Enigma, Ekiden does not allow on-chain computation but instead, the blockchain is solely used as a persistent state storage. Code and Private Inputs are provided by an off-chain client that exclusively communicates with enclaves. Once computation is finished, the enclave returns the Output directly to the client while the new state is checkpointed to the blockchain.

4.3 Secure Multiparty Computation-based Off-chaining

Secure Multiparty Computation (sMPC) protocols can be leveraged to design a privacy-preserving off-chain computation approach, which we describe in this section.

4.3.1 Concept.

SMPCs [33] enable a set of nodes to compute functions on secret data in a way that none of the nodes ever has access to the data in its entirety.

We leverage this to define a privacy-preserving off-chain computation scheme as introduced in Section 2.3 that computes a state transition based on secret data:

First, secret data is split into shares and the shares distributed to a set of Off-Chain Nodes as Private Inputs. Current state can be provided as Public Inputs. The off-chain nodes then compute the state transition for their share. Afterwards, they publish the Outputs that are then recombined to the final result and persisted on-chain. A key property that a sMPC protocol needs to fulfill to support this kind of off-chaining is public auditability [1], i.e., an *Auditor* not involved during the protocol can check correctness of the computation. Given this property, correctness can be checked by an on-chain Auditor during the Verify phase or an arbitrary off-chain Auditor by evaluating a computation's on-chain audit trail.

4.3.2 Realizations.

Enigma [34] proposed a privacy-preserving decentralized computation platform based on sMPCs where a blockchain stores a publicly verifiable audit trail. However, current sMPC protocols add too much overhead for such a network to be practical. Hence, Enigma now relies on Trusted Execution Environments as further explained in Section 4.2. Theoretical approaches based on Fully Homomorphic Encryption [16] involve even higher overheads and are not efficient enough to be practical.

4.4 Incentive-driven Off-chain Computing

In Incentive-driven Off-chain Computation (IOC) systems, incentive mechanisms are applied to motivate off-chain computation and guarantee computational correctness.

Table 1: Comparison of Off-chain Computation Approaches

		Scalability		*Privacy*	*Security*		*Programmability*
Approach	**Realization**	**On-chain Verification**	**Off-chain Computation**	**Private Inputs**	**Security Assumption**	**Post Quantum Security**	**Progr. Abstraction**
Verifiable Comput.	zkSNARKs [25]	*One-time setup:* $O(n)$, n number of multiplication gates in circuit *Repeated Verify Step:* $O(1)$ *Proof size:* 3 group elements [17], i.e., 127 bytes for BN128 curve	$O(n)$, n number of multiplication gates in circuit	yes	Knowledge of Exponent Assumption [15] & Trusted Setup was performed correctly	no	Arithmetic Circuits
	Bulletproofs [9]	*Verify:* $O(n)$, n number of multiplication gates in circuit *Proof size:* few kilobytes, $\log(n)$, n number of multiplication gates in circuit	$O(n)$, n number of multiplication gates in circuit	yes	Discrete Log Problem	no	Arithmetic Circuits
	zkSTARKs [2]	*Verify:* $O(\log T(n))$, n number of multiplication gates in circuit *Proof size:* few hundred kilobytes, $\log(n)$, n number of multiplication gates in circuit	$O(n)$, n number of multiplication gates in circuit	yes	Collision-resistant Hash Functions	yes	Higher Degree Polynomials
sMPC		*On-chain Auditor:* $O(n)$, n number of gates in circuit [1] *Off-chain Auditor:* None	$O(n)$, n number of gates in circuit	yes	At least one honest node & concrete protocol's assumptions	yes	Arithmetic Circuits
Enclave Systems		Validate enclave's attestation: $O(1)$, singature verification [18]	Native execution & attestation overhead	yes	TEE are isolated & Attestations cannot be corrupted & Trust in Attestation Certificates	no	Languages that compile into machine code executable by TEE
Incentive Systems		Binary search & one computation step: $O(\log(n))$, n number of computation steps [30].	Virtual Machine Overhead (Execution History)	no	Economic incentives lead to expected behaviour of network participants	yes	Languages that compile into VM instruction set used on- and off-chain

4.4.1 *Concept.*

IOC systems assume economical rational behavior such that participants strive to maximize their utility. System rules can be enforced by retaining deposits as a leverage against contravening activity and by financially rewarding desired behavior.

Referring to our off-chain computation model, computational tasks are redundantly executed by an off-chain *Solver* and multiple competing off-chain *Verifiers* that are incited to find errors in solutions published by Solvers. Public Inputs are published on-chain by the *Task Provider* as part of the task to be solved. If a Verifier disagrees with a Solver, the *Judge*, represented by the blockchain network itself, decides who is right and eligible to obtain the reward. Delivering the verdict represents the verification process. Once a correct Output is available it is publicly persisted to the State Storage.

4.4.2 *Realizations.*

IOCs inherit two critical design issues: (1) Keep Verifiers motivated to validate solutions and (2) reduce computational effort for the on-chain Judge. TrueBit [30], as the first IOC implementation, proposes solutions for both challenges. As Verifiers would stop validating if Solvers only published correct solutions, TrueBit enforces Solvers to provide erroneous solutions from time to time and offers a reward to the Verifiers for finding them. Further, the problem of the computationally limited Judge is solved by playing an interactive *Verification Game* [19] between Solver and Verifier. Applying binary search on the conflicting execution histories, the dispute that the Judge must decide on narrows down to a very small computational challenge which can be solved on-chain.

5 COMPARISON

A detailed comparison with regards to the properties of different off-chaining approaches is depicted in Table 1. In addition, we briefly elaborate on notable aspects in more detail in this section.

Scalability: As explained in Section 4.1.2, zkSNARKs require a setup phase which is more expensive than naive execution. After the setup, however, proof size and verification complexity are extremely small and independent of circuit complexity. This amortization makes zkSNARKs especially efficient for computations executed repeatedly, which is usually the case for off-chain state transitions. While zkSTARKs and Bulletproofs require no setup, proof size and verification complexity grow with circuit complexity, which limits applicability.

Security: While IOC systems like TrueBit [30] are safe under the assumption that there will always be at least one honest participant motivated by economic incentives, liveness can be impaired by malicious verifiers: They can challenge every computation step with erroneous solutions and hence enforce full on-chain execution.

EOC relies on TEE implementations such as Intel SGX which has been shown to be vulnerable [8, 14, 23, 28] and requires trust in Intel's Attestation Service for SGX [18].

6 CONCLUSION

In this paper, we introduced a generic off-chaining model describing different forms of off-chaining data or computation from the blockchain in order to enhance such systems.

Furthermore, we provided an overview of computation off-chaining techniques, described them, and compared them with regards to different desirable properties.

Future work could explore further off-chain computation techniques and study novel hybrid off-chaining approaches in detail as research progresses.

REFERENCES

[1] Carsten Baum, Ivan Damgård, and Claudio Orlandi. 2014. Publicly auditable secure multi-party computation. In *International Conference on Security and Cryptography for Networks*. Springer, 175–196.

[2] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. 2018. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046.

[3] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. 2015. Secure sampling of public parameters for succinct zero knowledge proofs. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 287–304.

[4] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. 2014. Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture.. In *USENIX Security Symposium*. 781–796.

[5] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. *CoRR* abs/1407.3561 (2014).

[6] Sean Bowe, Ariel Gabizon, and Matthew D Green. 2017. *A multi-party protocol for constructing the public parameters of the Pinocchio zk-SNARK*. Technical Report. TR 2017/602, IACR.

[7] Sean Bowe, Ariel Gabizon, and Ian Miers. 2017. Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model. Cryptology ePrint Archive, Report 2017/1050.

[8] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. 2017. Software Grand Exposure: SGX Cache Attacks Are Practical. *CoRR* abs/1702.07521 (2017).

[9] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy (SP)*. IEEE.

[10] Vitalik Buterin. 2014. Ethereum: A next-generation smart contract and decentralized application platform. *https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper* (2014).

[11] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah M. Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2018. Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contract Execution. *CoRR* abs/1804.05141 (2018).

[12] Jacob Eberhardt and Stefan Tai. 2017. On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In *European Conference on Service-Oriented and Cloud Computing*. Springer, 3–15.

[13] Jacob Eberhardt and Stefan Tai. 2018. ZoKrates - Scalable Privacy-Preserving Off-Chain Computations. In *IEEE International Conference on Blockchain*. IEEE.

[14] Jo Van Bulck et al. 2018. Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, Baltimore, MD, 991–1008.

[15] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. 2012. Quadratic Span Programs and Succinct NIZKs without PCPs. Cryptology ePrint Archive, Report 2012/215.

[16] Craig Gentry and Dan Boneh. 2009. *A fully homomorphic encryption scheme*. Vol. 20. Stanford University Stanford.

[17] Jens Groth. 2016. On the size of pairing-based non-interactive arguments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 305–326.

[18] Intel. 2013. Innovative Technology for CPU Based Attestation and Sealing. https://software.intel.com/en-us/articles/innovative-technology-for-cpu-based-attestation-and-sealing.

[19] Sanjay Jain, Prateek Saxena, Frank Stephan, and Jason Teutsch. 2016. How to verify computation with a rational network. *CoRR* abs/1606.05917 (2016).

[20] A Kosba, C Papamanthou, and E Shi. 2018. xJsnark: A Framework for Efficient Verifiable Computation. In *IEEE Symposium on Security and Privacy*.

[21] Brainbot Labs. 2017. Raiden Network. http://raiden.network/.

[22] Protocol Labs. 2017. Filecoin: A Decentralized Storage Network. *https://filecoin.io/filecoin.pdf* (2017).

[23] Ahmad Moghimi, Gorka Irazoqui, and Thomas Eisenbarth. 2017. CacheZoom: How SGX Amplifies The Power of Cache Attacks. *CoRR* abs/1703.06986 (2017).

[24] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[25] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. 2013. Pinocchio: Nearly practical verifiable computation. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 238–252.

[26] Joseph Poon and Vitalik Buterin. 2017. Plasma: Scalable autonomous smart contracts. *https://plasma.io/plasma.pdf* (2017).

[27] Joseph Poon and Thaddeus Dryja. 2015. The bitcoin lightning network: Scalable off-chain instant payments. https://lightning.network.

[28] Michael Schwarz, Samuel Weiser, Daniel Gruss, Clémentine Maurice, and Stefan Mangard. 2017. Malware Guard Extension: Using SGX to Conceal Cache Attacks. *CoRR* abs/1702.08719 (2017).

[29] Andrew Tam. 2018. Secret Voting Smart Contracts With Enigma: A Walkthrough. https://blog.enigma.co/secret-voting-smart-contracts-with-enigma-a-walkthrough-5bb976164753

[30] Jason Teutsch and Christian Reitwießner. 2017. A scalable verification solution for blockchains. (2017). https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf.

[31] Viktor Trón, Aron Fischer, Dániel A. Nagy, Zsolt Felföldi, and Nick Johnson. 2016. Swap, Swear and Swindle - Incentive System for Swarm. (2016).

[32] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper* (2014).

[33] Andrew C Yao. 1982. Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*. IEEE, 160–164.

[34] Guy Zyskind, Oz Nathan, and Alex Pentland. 2015. Enigma: Decentralized Computation Platform with Guaranteed Privacy. *CoRR* abs/1506.03471 (2015).