PRESERVING TRANSACTIONAL PRIVACY ON DISTRIBUTED LEDGERS USING ZERO-KNOWLEDGE PROOFS

Executive Summary

Since the emergence of distributed ledger technology (DLT), such as Blockchain, many have recognized the tremendous opportunity it presents and estimated the potential cost savings as well as the revenue from new service offerings in the tens of billions of dollars per annum. Yet despite all the apparent opportunities, clear ROI, and significant investment there have been very few production implementations of the technology to date.

One of the main reasons many DLT projects are not deployed in production is that they struggle with solving the inherent privacy problem in DLT. In a distributed ledger all transactions are visible to all the nodes. In most use cases this fact reveals sensitive business information, barring the technology from being used in production. Many approaches have been proposed to handle the privacy problems, such as circles of trust, channels, trusted execution environments, ring signatures, mixers, and stealth addresses. Each of these approaches have their drawbacks and limitations and, in most cases, do not provide a sufficient and practical solution.

Zero-knowledge proof is an emerging cryptographic technology that can provide complete transactional privacy. A zero-knowledge proof is a protocol by which one party can prove to another party that they know certain information without revealing the underlying data. Instead of revealing all the transaction details to all nodes, QEDIT generates a proof that the transaction is valid. Other nodes can verify the validity of the transaction without being exposed to the underlying transaction details. While generating zero-knowledge proofs is a resource intensive process, required transactions per second rates can be achieved by using QEDIT's patent-pending proof chaining technology. Proof chaining allows to break proofs to smaller proofs and run them in parallel to achieve the required throughput to support real-world use cases.

The Inherent Privacy Problem in Distributed Ledgers

Distributed ledger technologies (DLT), such as Blockchain, present a tremendous opportunity for enterprises and marketplaces. Many have recognized the potential of DLT to streamline business processes, slash operational costs, and enable new services by allowing business to collaborate in ways that until now were impossible due to lack of trust and transparency. Goldman Sachs estimated that the adoption of Blockchain could save $12 billion annually by streamlining clearing and settlement of cash securities [1]. J.P. Morgan estimated $10 billion in annual cost savings across the entire spectrum of front-to-back processes and operating metrics of an investment bank by significantly simplifying the correspondent banking structure using Blockchain [2]. A report by Spanish Bank Santander estimated that DLT could reduce banks' infrastructure costs attributable to cross-border payments, securities trading and regulatory compliance by $15-20 billion per annum by 2022 [3].

The promise of cost savings and new opportunities has prompted many businesses to make significant investments in DLT in recent years. A 2018 survey [4] conducted by Deloitte found that 39% of businesses are investing more than $5 million annually in Blockchain and 43% cited Blockchain as one of their top 5 strategic priorities. Clearly, organizations have identified the importance of this emerging technology and have taken concrete steps to leverage it in their business processes. Still, one may wonder how despite significant investment in DLT and the tremendous estimated ROI there are very few production implementations leveraging the technology to date. One answer is the inherent privacy problem in distributed ledgers.

In DLT every transaction is recorded in a public ledger that is shared between all the participants. Even in permissioned or private DLT implementations this quality of DLT presents a fundamental privacy problem for enterprises that completely bars the technology from being used in production. Let's consider a use case of a bank consortium that is looking to leverage DLT to streamline cross-border payment processing. Instead of using a chain of nostro accounts, the banks will be able to transfer payments directly. However, since all transactions are recorded in a ledger that is shared between all the banks confidential business information is exposed. When Bank A transfers money to Bank B all other banks can see the transaction details including who is the buyer, who is the seller, and the amount and asset type being transferred. In fact, every transaction of every bank is visible to all the other banks creating a fundamental roadblock for any production adoption of the technology. This glaring problem has led even prominent DLT proponents to concede that it is unlikely banks will adopt this technology in its current state due to privacy issues [5].

Some erroneously consider Blockchain and cryptocurrencies such as Bitcoin or Ethereum to be private since users are only identified using anonymous wallet IDs. It should be noted that although wallet IDs are anonymous, once someone transacts with a specific wallet ID, and having learned the identity behind the wallet ID, they can now trace all the transactions conducted by the specific person or organization. In fact, today there are several companies that offer this forensic analysis as a paid service to individuals, companies, and government agencies.

It is this inherent privacy problem that QEDIT solves by providing an enterprise solution that creates a privacy layer on top of DLT networks. QEDIT uses a cryptographic protocol called zero-knowledge proofs to deliver transactional privacy. Prior to zero-knowledge proofs, different approaches have been proposed over the years to preserve privacy over DLT networks. Each of these approaches has its drawbacks that make it impractical or

insufficient in most production use cases. We will survey these approaches and later discuss how zero-knowledge proof cryptography can be used to deliver complete transactional privacy on DLT.

Traditional Approaches for Handling the Privacy Problem

Circles of Trust / Channels

One of the first methods explored to deliver some level of privacy on Blockchain is encryption. Different Blockchain stacks refer to this method by different names but the mechanism is similar - establishing a private subnet of communication between two or more specific network nodes for the purpose of conducting confidential transactions. The transactions within the circle of trust are visible to all, but are encrypted from anyone outside the circle of trust. In this manner “sub-blockchains” are created on top of a main Blockchain network.

This approach presents two main challenges when attempting to perform token or asset transactions. The first challenge is how to transfer an asset to someone outside the circle of trust. Since this party does not have visibility to the entire transaction ledger inside the circle of trust they cannot verify that you actually own the asset and that you did not spend it previously. In order to show that this is indeed a valid transaction the sender needs to add the receiver to the circle of trust, exposing the receiver to sensitive transactional history. The second challenge is maintaining all the different encryption keys for any possible communication channel of 2 or more participants. The number of keys to generate, coordinate, and maintain grows exponentially as more nodes are added to the network quickly becoming impractical to scale.

Trusted Execution Environment

Another solution to transacting privately on DLT uses trusted hardware in the machine running the transaction. One such example is Intel’s SGX which provides an isolated environment called an “enclave”, in which computations can be ran and signed using cryptographic keys that are embedded within the secure hardware. The enclave signs both the result and the code that was executed. The verifier simply needs to check that the result was signed with the proper keys.

The risk in using trusted execution environments is that they are prone to hacking. A hacker can get access to the hardware component and work offline to identify weaknesses in its security until an attack is discovered. For example, in 2018 two security issues were discovered in Intel’s SGX trusted execution enclave. In February 2018 a team of researchers from Ohio State University published a paper [6] revealing an attack called SgxSpectre that enables an attacker to steal the keys signed by Intel from the SGX enclave. This would allow an attacker to forge transactions, double spend, or issue new tokens with no restrictions.

In August 2018 a second team of researchers published another paper [7] demonstrating an attack called Foreshadow that enables extracting the full cryptographic keys from Intel’s SGX enclave which can be used to arbitrary forge local and remote attestation responses. These attacks follow a string of publications in 2017 that demonstrated numerous other vulnerabilities in SGX [8] [9] [10]

Obfuscation and Decoy Methods

One of the most prevalent attempts at providing transactional privacy on DLT is using different methods to obfuscate the data or to mix the transaction with other decoy transactions. These methods include ring signatures, mixers, ring confidential transactions, stealth addresses, deterministic wallets, CoinJoin and many others. These are very different techniques that achieve a similar result - obscure one or more of the transaction details such as the sender, the receiver, or the amount paid.

The main problem with using these methods is that some data and meta data about each transaction are still leaked and over time this information leaks can be used to reveal transaction details as well as trace the transaction history and future transactions of a particular asset. At the 2018 Devcon conference Dr. Ian Miers of Cornell Tech, an expert in computer security and applied cryptography, presented three different attacks on obfuscation and decoy methods that reveal the transaction details [11]. While a single transaction may be kept private using one or more of these methods, Dr. Miers shows how it is quite easy to reveal this information when there are repeated transactions between the participants. This problem is amplified in most permissioned Blockchain or enterprise DLT use cases since the participants repeatedly interact with each other and generally there are far fewer participants than in public Blockchain networks.

Introduction to Zero-Knowledge Proofs

The advanced method QEDIT uses to completely preserve transactional privacy is called zero-knowledge proofs. Zero-knowledge proofs is a cryptographic method that was invented in the 1980's but only recently has become practical. A zero-knowledge proof is a protocol by which one party can prove to another party that they know certain information without revealing the underlying data. While it is trivial to prove this knowledge by simply sharing the underlying data, zero-knowledge proofs enable doing so without the sharing the data. As is described in the next section, this powerful protocol can be applied to create private transactions on a DLT network.

A zero-knowledge proof must satisfy three properties -

- Completeness - For a true statement, an honest verifier should be able to be completely convinced the statement is true by a proof provided by an honest prover
- Soundness - For a false statement, a dishonest prover should not be able to convince a verifier that the statement is true.
- Zero-Knowledge - The verifier does not learn anything other than the fact that the statement is true.

The mathematics behind zero-knowledge proofs are quite complex but the concept can be easily explained using an example. There is a famous series of children's' books called "Where's Wally?" (or "Where's Waldo" in the United States). In these books children are challenged to find a character named Wally who is hidden in a colorful page containing many other drawn characters. Suppose that a child is unable to find Wally despite significant efforts, becomes frustrated and begins questioning whether Wally is actually there in the page. You can prove to the child Wally is there by simply pointing to him in the picture, but that would also solve the puzzle and eliminate the fun. So how can one prove to the child that Wally is in the page?

You can take a large envelope about 4 times the size of the book, cut a small hole in the envelope the size of Wally, and then place the book within the envelope in a way that only Wally is seen through the hole in the envelope. You show the child that Wally can be seen through the hole. At this point you seal the envelope, shake it, and hand it to the child. The child is now satisfied that Wally is indeed in the page, but they cannot retrace his location in the page. This process satisfies the three properties of a zero-knowledge proof - If the page contains Wally, the child is convinced of that fact, if the page does not contain Wally, it is impossible to convince the child otherwise, and the child learned nothing in the process other than the fact the Wally is in the page.

Zero Knowledge Proofs were co-invented in 1989 by Prof. Shafi Goldwasser [12] who won the Turing Award for her groundbreaking work in 2012. One of the first practical implementations of a general purpose zero-knowledge proof scheme is called zk-SNARK (Zero Knowledge Succinct Non-interactive Arguments of Knowledge) and was pioneered by cryptography expert Prof. Eran Tromer [13] [14] [15]. Both Prof. Goldwasser and Prof. Tromer sit on QEDIT's scientific advisory board.

Transactional Privacy using Zero Knowledge Proofs

In a regular DLT transaction, nodes verify and confirm a transaction by simply looking at the details and verifying the sender indeed owns the asset and there is no double spend. In contrast, an asset transfer performed by QEDIT does not reveal the details of the transaction yet still allows other nodes to verify and confirm the validity of the transaction by providing a zero-knowledge proof that the transaction is valid.

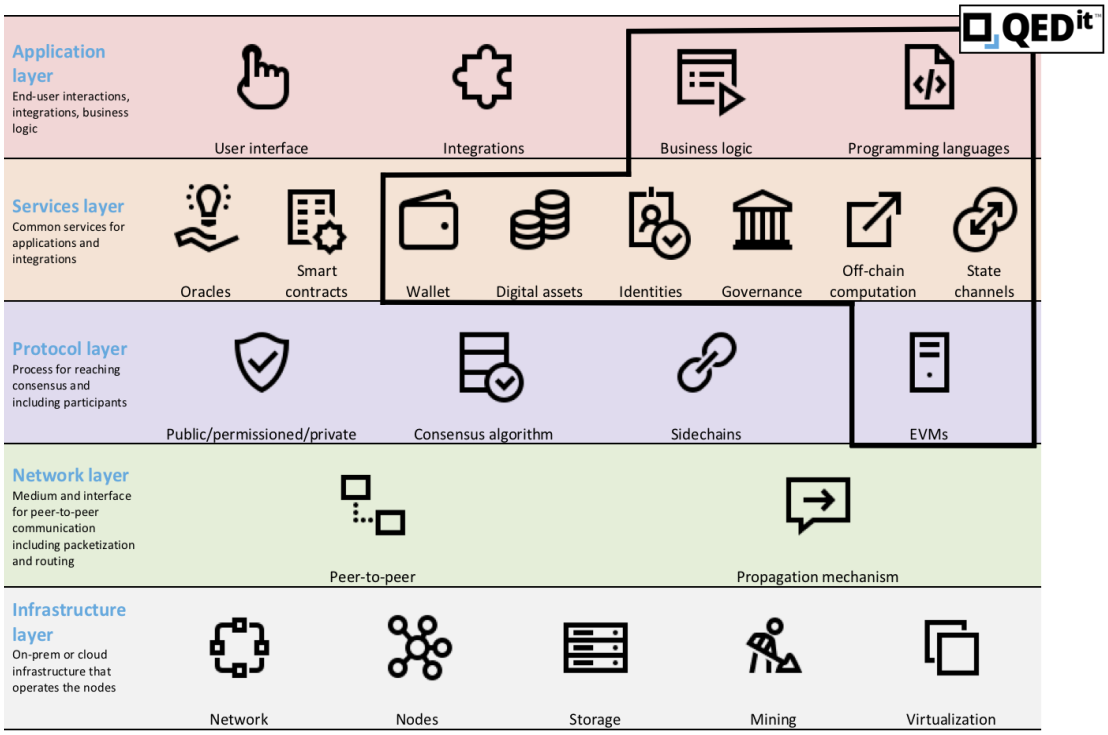
When ownership of an asset is transferred QEDIT provides a protocol that uses zero-knowledge proofs to attest to the validity of the transaction. The protocol ensures that:

- The asset indeed exists and is owned by the sender.
- The sender is authorized to transfer the ownership.
- There was no double spending of the asset.
- The total amount of the asset was preserved as part of the transaction and no new assets were created in the process.
- The ownership was transferred to the new owner.
- The type of the asset was maintained in the transfer process. The proof needs to ensure that if the owner is transferring an asset of type 1, then the receiver did indeed receive an asset of type 1. Since QEDIT supports multiple asset type on the same network, and assets can be either fungible or non-fungible this confirmation is important to ensure the validity of the transaction.
- All custom business logic that is defined to run as part of a transaction was executed successfully. In addition to the intrinsic transfer logic, QEDIT allows customers to define their own custom business logic that must run as part of a transaction. This can be done in order to ensure the transaction conforms to regulations such as KYC or AML. As such, as part of the transaction proof, QEDIT includes a confirmation that this custom business logic was followed.
- The DLT public state was updated.

The proof is constructed as part of the transaction action and published to the DLT. Other nodes confirm the transaction by verifying the proof. By this verification process,

nodes can be certain that the transaction is valid although they do not know who is the seller, who is the buyer, and what they are selling, or any other metadata about the transaction.

Zero-knowledge proofs are constructed from low-level logical gates. Understanding how to construct large complex gates schemes to securely prove real-world queries and constraints requires great expertise in cryptography. QEDIT provides client SDKs in several common high-level languages that make it simple to make transactions without the need to build these complex gates constructions.



The Blockchain stack and the services and functionality provided by QEDIT.

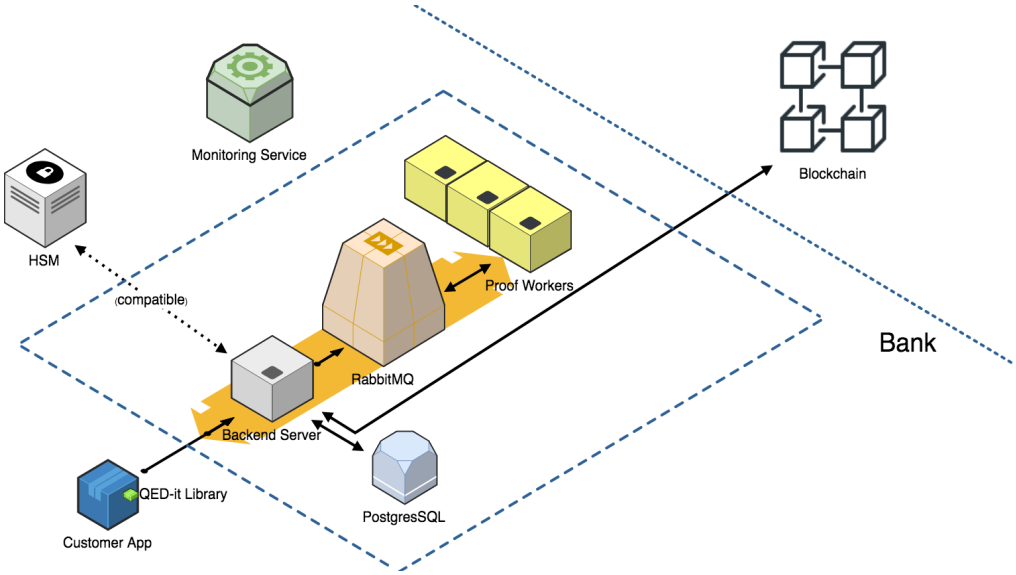
The QEDIT solution provides additional important services and functionality that span across several Blockchain stack layers such as wallets, issuance of digital assets, custom business logic, and shared public state that is based on proofs and hashed commitments rather than transactional data. These services make it simple to build feature-rich applications on the Blockchain stack that preserve data privacy.

Achieving Scalability and Robustness through Proof Chaining

Constructing zero-knowledge proofs is a resource intensive process. The running time of this construction greatly depends on what is being proven, but generally can take anywhere from one to several seconds. For example, the Sapling protocol developed by the ZCash cryptocurrency team completes a single transaction proof in 2.3 seconds [16]. Yet some DLT use cases can require a higher transactions per second rate. In addition, the more information is being proven the greater and greater is the memory requirements for the machine generating the proof.

To handle these challenges QEDIT developed a patent-pending process called proof chaining with which large proofs can be broken down to smaller proofs that are ran in

parallel. The smaller proofs are then “glued” together at the end to provide verification of the integrity of the entire proof. The smaller proofs are generated on stateless “proof worker” server instances. This allows elastic scale up or down of proof worker servers in real time depending on the current workload. In this manner it is possible to reach a much higher transactions per second rate. In addition, the overall memory requirements for each proof worker is also much lower than for generating a single large proof.



QEDIT enterprise solution with a parallel proof workers autoscaling group

Another added benefit to using proof chaining is the modularity it enables to use different zero-knowledge proving schemes. There are different proving schemes such as zk-SNARK, bulletproof, and zk-STARK.

	Proof size	Proving time	Verification time
zk-SNARK	●	●	●
zk-STARK	●	●	●
Bulletproof	●	●	●

Zero-knowledge proving schemes comparison
Source: Demystifying Zero Knowledge Proofs [17]

As can be seen in the table above each scheme has different strengths and drawbacks. By breaking down large proofs to smaller proofs, it is possible to use different proving schemes for each smaller proof. In this way the large proof is optimized by applying the most appropriate schemes available today.

Proof chaining is a powerful process that allows QEDIT to generate complex zero-knowledge proofs to support real-world use cases. The parallelization of the proving processes makes it practical to enforce custom business rules as part of the transaction while still maintaining desired transactions per second rates. Moreover, since each proof worker is stateless the QEDIT solution delivers the robustness and scalability enterprises require for production usage.

Standardizing Zero-Knowledge Proofs

While the academic work on zero-knowledge proofs has been ongoing since its invention in the 1980's, it is only in recent years that different practical implementations of different general-purpose proving scheme have been made available. Indeed, it seems the rate of innovations of zero-knowledge proof technology is at its all-time high. The proliferation of zero-knowledge proofs created a need for industry standards to provide interoperability, performance benchmarks, and protocol security. Having identified this need, QEDIT is leading an open initiative by industry and academic leaders to standardize the use of zero-knowledge proofs called zkproof.org. The effort was launched in a workshop at MIT in May 2018 and drew a cross industry and academia list of participants including from Microsoft, IBM, R3, VMware, Stanford University, MIT, Deloitte, EY, BNP Paribas, and ING Bank.

The standardization effort includes 3 main tracks. The first is the security track. The goal of this effort is to define security standards for zero-knowledge proofs and a standard process for certifying different proving schemes. The second track is the implementation track which focuses on enabling interoperability by building a standard interface that application developers can use to interact with zero-knowledge proof systems. The last track is the applications track. This track reviews common use cases where zero-knowledge proof is applied such as asset transfers.

The zkproof.org standardization effort is an important step towards defining clear guidelines, standards, and performance reviews of various zero-knowledge proof schemes and applications, fostering clarity and trust in this emerging technology. As such, QEDIT is committed to continue leading and supporting this effort and to adopt the community guidelines and standards produced by the effort.

Conclusion

It is a paradox that the greatest value proposition of DLT, transparent collaboration between organizations and individuals, also poses an inherent privacy problem. Solving the inherent privacy problem in distributed ledgers is crucial for unlocking the full potential of the technology. Many efforts have been taken to date to provide privacy on DLT the culmination of which is using zero-knowledge proof cryptography. Constructing zero-knowledge proofs to handle real use cases and business requirements requires great expertise in cryptography. QEDIT provides a scalable and robust enterprise solution for preserving transactional privacy on DLT that abstracts and automates the complex proofs constructions. Integration of QEDIT's solution to existing projects is made easy through high-level language client SDKs. Finally, while generating zero-knowledge proof is a resource intensive process, achieving required throughput rates to support production use cases is possible using QEDIT's patent-pending proof chaining technology.

References

[1] Blockchain Putting Theory into Practice. 2016. Report by Goldman Sachs
https://github.com/bellaj/Blockchain/blob/master/Goldman-Sachs-report-Blockchain-Putting-Theory-into-Practice.pdf

[2] Blockchain and the decentralization revolution. 2018. Report by J. P. Morgan
https://www.jpmorganchina.com.cn/jpmpdf/1320745566550.pdf

[3] The Fintech 2.0 Paper: rebooting financial services. 2015. Report by Santander InnoVentures, Oliver Wyman, and Anthemis
http://santanderinnoventures.com/wp-content/uploads/2015/06/The-Fintech-2-0-Paper.pdf

[4] Breaking blockchain open Deloitte's 2018 global blockchain survey. 2018. Report by Deloitte
https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-2018-global-blockchain-survey-report.pdf

[5] Anna Irerra. Banks unlikely to process payments with distributed ledgers for now, says Ripple. 2018. Article in Reuters
https://www.reuters.com/article/us-blockchain-ripple/banks-unlikely-to-process-payments-with-distributed-ledgers-for-now-says-ripple-idUSKBN1J92JG

[6] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, Ten H. Lai.
SgxPectre Attacks: Stealing Intel Secrets from SGX Enclaves via Speculative Execution. 2018.
https://arxiv.org/abs/1802.09085

[7] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, and Raoul Strackx. FORESHADOW: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. 2018.
https://foreshadowattack.eu/foreshadow.pdf

[8] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel and Tilo Müller. Cache Attacks on Intel SGX. 2017.
http://www.sharcs-project.eu/m/documents/papers/a02-gotzfried.pdf

[9] Yeongjin Jang, Jaehyuk Lee, Sangho Lee and Taesoo Kim. SGX-Bomb: Locking Down the Processor via Rowhammer Attack. 2017.
https://taesoo.kim/pubs/2017/jang:sgx-bomb.pdf

[10] Michael Schwarz, Samuel Weiser, Daniel Gruss, Clémentine Maurice and Stefan Mangard. Malware Guard Extension: Using SGX to Conceal Cache Attacks. 2017.
https://arxiv.org/abs/1702.08719

[11] Ian Miers. Satoshi Has No Clothes: Failures in On-Chain Privacy. 2018. Presentation at DevCon4.
https://www.youtube.com/watch?time_continue=3&v=9s3EbSKDA3o

[12] Shafi Goldwasser, Silvio Micali and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. 1989.
https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Proof%20Systems/The_Knowledge_Complexity_Of_Interactive_Proof_Systems.pdf

[13] Nir Bitansky, Ran Canetti, Alessandro Chiesa and Eran Tromer. From Extractable Collision Resistance to Succinct Non-Interactive Arguments of Knowledge, and Back Again. 2011. https://eprint.iacr.org/2011/443

[14] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer and Madars Virza. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge. 2013. https://eprint.iacr.org/2013/507

[15] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. 2014. https://eprint.iacr.org/2014/349

[16] Paige Peterson. Reducing Shielded Proving Time in Sapling. 2018.
https://z.cash/blog/reducing-shielded-proving-time-in-sapling/

[17] Elena Nadolinski. Demystifying Zero Knowledge Proofs. 2018.
https://docs.google.com/presentation/d/1gfB6WZMvM9mmDKofFibIgsyYShdf0RV_Y8TLz3k1Ls0/edit#slide=id.g443ebc39b4_0_92

About QEDIT

Comprised of world-class entrepreneurs, researchers and developers, QEDIT provides the industry's first enterprise solution for preserving privacy over Blockchain networks. By applying advanced zero-knowledge proof cryptography, QEDIT enables financial institutions to unlock the full potential of the Blockchain. QEDIT provides several native language SDK's for interacting with the QEDIT solution accelerating Blockchain development and production roll-out for the world's largest organizations.

www.qed-it.com | info@qed-it.com | Ehad Ha'am 54 Tel Aviv, Israel