Version 1.1 / 5 October 2020

Qredo Network

Brian Spector[1]*

[1]PPMC Member and Contributor to Apache Milagro, Chief Product and Strategy O°cer at Qredo Ltd

Correspondence
Qredo Ltd, Kemp House, 152 - 160 City Road, London, UK, EC1V 2NX
Email: info@qredo.com

The Qredo Network enables the financialization of crypto assets; instant trading, atomic swaps and lending happens within a secure, hardware hardened decentralized network. This paper provides detail over the cryptographic protocols and implementation methods that enable its functional operations.

1 | INTRODUCTION

In the 'wild west' of the new digital asset economy, traders face a series of risky technical challenges.

Assets must be shuffled between hot wallets—which sacrifice security for liquidity, and cold wallets—which forfeit fast settlement times for safety. This treacherous trade-off means digital assets are either vulnerable to theft, but readily available for immediate trading opportunities, or relatively secure, but stuck in cumbersome wallets which prolong settlement times.

Wherever they are kept, crypto assets are likely to be commingled; making full transparency impossible and deterring large institutions from investing.

1.1 | Sector Irony

The irony of the situation is that by straying from its principles and centralizing custodian operations, the blockchain industry has degraded its own growth prospects. Institutional investors, who have the deepest pockets and the most ability to take crypto assets mainstream, cannot tolerate the risks of centralized custody.

But if decentralization is applied to custodial operations, then the requirements of fiduciary institutional investors can be met.

Qredo's Layer 2 digital asset tracking and settlement infrastructure removes the risks of dealing in digital assets with cryptographically provable security that enables real-time crypto asset transfer, atomic swaps and lending.

*With assistance from Qredo Ltd staff and other Apache Milagro contributors.

2 | A NEW APPROACH: DECENTRALIZED CRYPTO ASSET TRACKING

The Qredo Network is a decentralized approach to crypto asset safekeeping, tracking, deliver and settlement. The consequence of decentralization is that secured transfer, lending or atomically swapping crypto assets between parties is realizable via a decentralized settlement network. In essence, the network becomes the vault.

The idea is straightforward: A distributed ledger is used to record the ownership of a crypto asset. For each crypto asset, there is a corresponding set of clients who generate recorded digital signatures that enable a quorum of a class of computers, called MPC Nodes, to run a multi-party computation (MPC) protocol. Each class of MPC Node (Client or Server) has its own set of secured secrets.

The MPC Nodes running the MPC protocol can create a public address for crypto assets to be deposited, and a signature on a transaction recognized by the underlying blockchain (example: Bitcoin) to spend or move the crypto assets from that public wallet address. When consensus determines that an actor 'owns' a crypto asset as stated by the Qredo Network blockchain, the asset owner is enabled, via the Qredo Core Client (Qredo's blockchain client software), to invoke a turn of the MPC protocol run by dedicated MPC Nodes within the Qredo Network.

Qredo Core Client software possesses a secret only it knows which is necessary for the invocation of the MPC protocol between the MPC Nodes, and the MPC Nodes possess their own secrets for running the protocol pertaining to that specific crypto asset. The MPC Nodes within the Qredo Network use the consensus derived determination of who owns what crypto assets to accept (or reject) invocations from Qredo Core Clients.

Lastly, a threshold decryption scheme run by a subset of MPC Nodes (in the same class), working in collaboration, decrypts their secrets that are necessary to run their end of the MPC protocol. They will engage in this protocol only if the request is deemed legitimate via their own interrogations of the blockchain.

The Qredo Network is built on the Tendermint protocol[1]. Tendermint is a general purpose blockchain consensus engine that can host arbitrary states of deterministic applications. Tendermint is software for securely and consistently replicating an application on many machines. Non-faulty machines see the same transaction log and computes the same state.

Tendermint consists of two chief technical components: a blockchain consensus engine and a generic application interface. The consensus engine, called Tendermint Core, ensures that the same transactions are recorded on every machine in the same order. The application interface, called the Application BlockChain Interface (ABCI), enables the transactions to be processed in any programming language.

There are three entities running different programs within the Qredo Network: Clients, Validators and MPC Nodes.

The cryptographic code within the Qredo Core Client comes from Apache Milagro, an open source software project under the Apache Software Foundation. The project's website and software is hosted and published by the Apache Software Foundation at `https://milagro.apache.org`. The crypto engines from Apache Milagro are well regarded, and are in use within some of the most prominent blockchain projects today, including Hyperledger. The core team responsible for contributing and maintaining the Apache Milagro crypto libraries are the developers of the Qredo Network.

Validator Nodes are responsible for committing new blocks in the blockchain. These validators participate in the consensus protocol by broadcasting votes which contain cryptographic signatures signed by each validator's private key. Validator Node software is published by the open source Tendermint project.

MPC Nodes work collectively together to provide a public key and signature functionality without every possessing or materializing a private key. The cryptographic code within the MPC Nodes also comes from Apache Milagro,

[1] Tendermint Protocol: https://tendermint.com

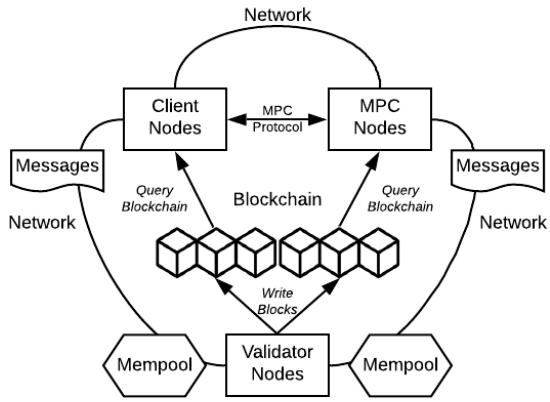

FIGURE 1 Qredo Network

an open source software project under the Apache Software Foundation.

Functionally, the Qredo Network:

1. **Eliminates Theft**: Operators of the Qredo Core Client ("Principals") can invoke the MPC Nodes to create public wallet addresses that have no known and never manifested private key, eliminating the burden of securing hot wallets and cold storage against private key theft. This happens by the employment of a multi-party computation (MPC) protocol run between the MPC Nodes.

The MPC protocol is a threshold signature scheme[17] enabling n parties to share the power to issue digital signatures under a single public key. A threshold t is specified such that any subset of $t+1$ players can jointly sign, but any smaller subset cannot. The protocol produces signatures that are compatible with an existing centralized signature scheme, such as ECDSA. The ECDSA key generation and signature algorithm are replaced by a communication protocol between the parties, but the ECDSA verification algorithm remains identical to the verification of an ECDSA signature as if it was created by a centralized party with access to the private key.

The goal of this design is that attacks to recover the private key near the effort of brute force guessing, reducing the chances of success to impossible odds. The design serves an additional purpose: It enables the near instant transfer of the right to run the MPC protocol using the Qredo Network blockchain. In essence, this transfers ownership of the asset at near real-time speed between counterparties on the Qredo Network.

2. **Hardware Hardened Blockchain**: Each instantiation of the MPC Node and Validator Node software happens on a specialized computing environment ("appliance") with integrated Hardware Security Modules (HSMs) and Secure Element (SE).

An HSM is a physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocessing. These modules traditionally come in the form of a plug-in card or an external device that attaches directly to a computer or network server. HSMs are tested against requirements found in FIPS 140-2, Security Requirements for Cryptographic Modules standards created by NIST. Security requirements cover 11 areas

related to the design and implementation of a cryptographic module. For each area, a cryptographic module receives a security level rating (1-4, from lowest to highest) depending on what requirements are met. The hardware utilized Qredo Network appliances is rated at Level 3.

A Secure Element (SE) is a microprocessor chip which can store sensitive data and run programs securely like the kind found in credit card payment systems. It acts as a vault, protecting what's inside the SE (applications and data) from malware attacks that are typical in the host (i.e. the device operating system) but also from the surrounding physical environment. Qredo Network appliances use a Secure Element to tamper proof its physical environment. An unauthorized attempt to access the internals of a Qredo Network appliance will instantly destroy the cryptographic keys stored within. The cryptographic keys and primitives required for operating the D-TA and Validator Node in most instances are created within and never leave the boundaries of the FIPS 140-2 Level 3 rated HSMs or Secure Element on the hardware appliances they are operating on. Where cryptographic primitives reside on disk, the keys to secure those primitives (AES keys, etc.) are created within the boundary of secured hardware and never leave the boundary.

3. **Governance**: In the Qredo Network, the ownership of the crypto asset is recorded on the Qredo Network distributed ledger. A **Transaction Right** (TR) is a transferable privilege that belongs to the owner of a crypto asset. Ownership of a crypto asset/TR enables a Qredo Core Client to unlock and invoke the MPC protocol between the MPC Nodes to create a cryptocurrency transaction on the crypto asset's underlying blockchain (example: Bitcoin). MPC Secrets are the decrypted archive of cryptographic primitives an MPC Node on Qredo Network needs to possess in order run the MPC protocol. Ownership of crypto assets is determined by consensus, and the rules that determine ownership relate to the aggregation of digital signatures specifying ownership, and approving transfer, atomically swapping assets or some other action. These types of digital signature are called BLS signatures, the details of which are described in 4.1. These BLS signatures, and the public keys that verify them, can be aggregated to achieve a complete single signature from a collection of individual signatures[5], with individual signatures themselves consisting of a subgroup of a threshold of signatures[2]. This is possible provided the correct Proof of Possession safeguards are in place against rogue public-key attack or the particular protocol instance provides its own defense[3].

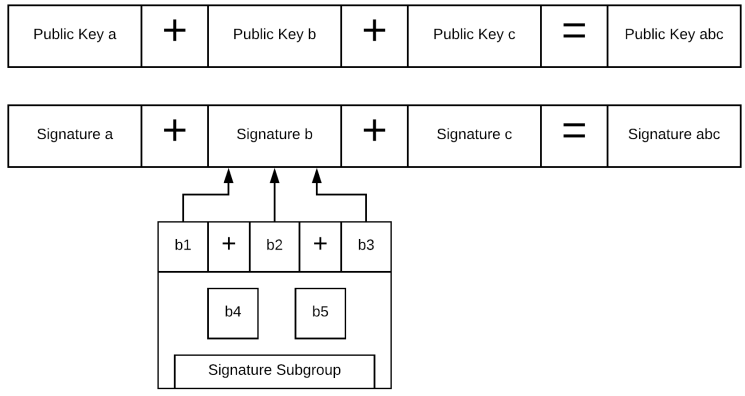

FIGURE 2 BLS Signatures

The digital signatures also act as an immutable audit record of which entities ("Custodians") approved (by way of a BLS signature they create) requests to approve the transfer, lending, atomic swap or other action pertaining to a

crypto asset between counterparties on the Qredo Network. These signatures are recorded into the Qredo Network's blockchain. Custodians can be organized in subsets and thresholds that map to the specific business organizational requirements. Each individual Custodian who provides a BLS digital signature in a subgroup can have their signature verified as providing a part necessary for the subgroup's aggregated signature to complete.

4. **Near-Instant Transfers and Atomic Swaps**: A Qredo Core Client may invoke the MPC protocol to obtain a public key and have the public key signed using the MPC protocol's signature capability. This signature is identical to a signature that would have been created with a private key, were it to exist (it does not).

BLS Signature aggregation is employed to record the ownership and transfer of crypto assets. Validator Nodes prevent the double transfer or double swap of any crypto assets through the record of aggregated BLS signatures in the blockchain. To transfer the crypto asset, a Principal with ownership of a crypto asset creates an aggregated public key from its public key and that of the Beneficiary to whom it is transferring the crypto asset. It creates a transfer message noting an Asset Identifier, its identity and that of the Beneficiary. Imagine a crypto asset contained in a wallet with 500 BTC, the ID beginning with '5a4...' is being transferred from Principal X to Beneficiary Y. Example:

Asset ID 5a4... transfer from Principal ID (x) to Beneficiary ID (y).

Principal X signs this message with its BLS secret key and submits it to the network. Once confirmed on the blockchain, the Beneficiary's client sees the message and acts on it. Assuming it agrees to the transfer, Beneficiary Y signs a duplicate of the message, which also confirms into the blockchain. Qredo Network Validators act on these confirmed messages by aggregating the signatures of the Beneficiary and Principal together. If the aggregated signature (created by the Validators) is verified by the aggregated public key (created by the Principal transferring the crypto asset), the Transaction Right, which enables the MPC protocol to run to signature completion, can only be run by the Beneficiary Y, as the Transaction Right, according to the blockchain, now belongs with Beneficiary Y, the owner of the crypto asset.

Imagine a scenario where a two counter-parties X and Y which to atomically swap their respective crypto assets. To enable a swap, an example transfer message would be:

Asset ID 5a4... transfer from Principal ID (x) to Beneficiary ID (y) AND Asset ID 698... transfer from Principal ID y to Beneficiary ID x

The same public key and signature aggregation as described above holds, but now with the Validators recognizing both crypto asset transfers (i.e., an atomic swap).

5. **Privacy**: All entries on the Qredo Network's blockchain have only references to an 'Account Code', a unique, non-personally identifiable ID of actors on the system, and an 'Asset ID', which relates to the underlying crypto asset, but reveals no information on-chain about what that crypto asset actually is. All messages not expressly bound to the transfer of crypto assets and the Custodian approval process are kept off chain and sent via the Matrix protocol[2]. Matrix is accurately described as a decentralised conversation store, rather than a messaging protocol. When sending a message in Matrix, it is replicated over all the servers whose users are participating in a given conversation - similarly to how commits are replicated between Git repositories. Matrix provides state-of-the-art end-to-end-encryption via the Olm and Megolm cryptographic ratchets. This ensures that only the intended recipients can ever decrypt

[2] Matrix Protocol: https://matrix.org

messages, while warning if any unexpected devices are added to a conversation. Matrix's encryption is based on the Double Ratchet Algorithm popularised by Signal, but extended to support encryption to rooms containing thousands of devices. Olm and Megolm are specified as an open standard and implementations are released under the Apache license, independently audited by NCC Group.[3]. Additionally, counterparties are verified to themselves using the Qredo Network's Encrypted Envelope messaging format as described in Section 4.5.

2.1 | ID Document

Each Qredo Core Client that connects to the network creates a self-sovereign identity document (ID Document) upon initialization; a random seed value is created at initialization which serves as the deterministic function to create other key pairs and primitives.

From the seed value, the following key pairs and primitives are created:

- BLS Public/Private Key Pair
- SIKE/RSA Public/Private Key Pair

Seed values which generate the key pairs are protected inside Hardware Security Modules (HSMs), such as a Yubikey. Note that all keys and cryptographic primitives necessary for Validator Nodes or MPC Nodes are protected with FIPS 140-2/3 HSMs, with the actual enclosures protected via tamper proof systems anchored to a secure element.

The public/private key pairs are signed by the private keys and the public keys and signatures are listed in a set in the JSON formatted ID Document which is written by the software client into Qredo Network's blockchain. The SHA-256 hash of the ID Document becomes the **Account Code** of the Qredo Core Client.

2.2 | Principals, Beneficiaries, Custodians

Principals: These entities operate a Qredo Core Client to connect to the Qredo Network. The Principal instructs Qredo Core Client to create a Fund with Assets and assign a Custodian policy over the Fund. The Custodian policy is a digitally signed document which states which Custodians have governance responsibilities over the Fund, and therefore must approve any transfer, atomic swap or settlement to the underlying blockchain of assets out of this Fund Additionally, the Principal may delegate authority to other Principals to invoke a transfer, atomic swap or settlement of assets into or out of the Fund.

Custodians: These entities operate a Qredo Core Client to connect to the Qredo Network. These entities work in coordinated steps to generate digital signatures over messages confirming their approval over a transfer, atomic swap or settlement of a crypto asset belonging to a Fund over which they have been appointed Custodian with governance responsibilities. The Principal's Qredo Core Client as a Master Custodian, organizing the communications with Custodians, collecting the signatures via the Matrix communication protocol, running a signature combiner function and finally submitted the message with aggregated signatures to the Qredo Network blockchain. The final aggregated/combined BLS signature is a critical step in the cryptographic scheme that results in the evidence of transfers and approvals on-chain.

Beneficiary: These entities operate a Qredo Core Client to connect to the Qredo Network. These entities receive

[3] https://www.nccgroup.trust/us/our-research/matrix-olm-cryptographic-review/

a transfer of a crypto asset either from a one-way transfer, atomic swap, lending facility or sweep of asset based upon a creditor arrangement.

3 | CRYPTOGRAPHY

3.1 | Pairing Cryptography

Pairing Based Cryptography has been used to provide solutions to intractable problems such as identity based encryption. The primitive that is used in the Qredo Network is the Type-3 pairing [15].

A Type-3 pairing is a mapping $_1 \times _2 \rightarrow _T$. The groups $_1$, $_2$ and $_T$ are all of the same prime order q. A pairing works on a special pairing-friendly elliptic curve [14].

The pairing is a function of two inputs, $P \in _1$, $Q \in _2$, such that the output $C \in _T$

$$C = e(P, Q)$$

The bilinearity of the pairing is the key characteristic that makes pairing interesting and widely used in cryptography.

$$e(aP, Q) = e(P, Q)^a = e(P, aQ)$$

The Qredo Network curve choice is BLS12-381 which provides security at the AES-128 level[8].

The first use of pairings in cryptography was in 1991, when they were used to attack certain elliptic curve cryptosystems that used supersingular elliptic curves. Between 2001 and 2004, three seminal papers used pairings in a constructive manner to implement novel (or vastly improved) protocols: Boneh and Franklin's identity-based encryption scheme[4], Boneh, Lynn, and Schacham's short signature scheme ("BLS signatures" for short)[7], and Joux's one round tripartite key exchange[18]. Their work jump-started interest in pairing-based cryptography, which has grown exponentially since then. By 2004, there were already over 200 articles published on this topic, and the number today is in the thousands.

Today, pairing-based cryptography is a well understood art that has found particular relevance within decentralized networks. Vitalik Buterin (of Ethereum) has written extensively on the subject[10]. In signature schemes, BLS signatures are widely recognized within the cryptocurrency space for their signature aggregation abilities. BLS signatures are now going through an IETF submission review[6] standardization process. Apache Milagro's pairing-based zero-knowledge proof multi-factor authentication protocol (ZKP MFA)[20][21][22], is widely deployed across cloud infrastructures and in public-facing deployments by the UK government[1].

Zcash implements their own pairing-based zero-knowledge proof algorithm named zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge)[9]. zk-SNARKs are used for protecting privacy of transactions of Zcash. Pairings are a key ingredient for constructing zk-SNARKS. Cloudflare introduced Geo Key Manager to restrict distribution of customers' private keys to the subset of their data centers. To achieve this functionality, attribute-based encryption is used and pairings again are a key building block. The Trusted Computing Group (TCG) specifies ECDAA (Elliptic Curve Direct Anonymous Attestation) in the specification of Trusted Platform Module. ECDAA is a protocol

for proving the attestation held by a Trusted Platform Module (TPM) to a verifier without revealing the attestation held by that TPM. Pairing cryptography is used for constructing ECDAA. FIDO Alliance and W3C have also published ECDAA algorithms similar to TCG.

The Qredo Network makes extensive use of BLS signatures as detailed in Section 4.1.

3.2 | Paillier Cryptosystem

The Paillier Cryptosystem[4] is a public key cryptosystem based on integer factorization and the DLOG problem. It encrypts plaintexts in $\mathbf{Z}_n$ to cyphertexts in $\mathbf{Z}_{n^2}$, where n is of the form pq for p and q primes. Moreover, it is an additive and multiplicative homomorphic cryptosystem. In particular for any $a, b, c \in \mathbf{Z}_n$ it holds that $E(a) \cdot E(b) = E(a+b) \bmod n^2$ and $E(a)^c = E(ac) \bmod n^2$.
The scheme uses the integer factorization trapdoor to generate keypairs $SK = \lambda = \phi(n) = (p-1)(q-1)$ and $PK = (g, n)$, where $g \in \mathbf{Z}_n^2$ has an element with order k a nonzero multiple of n. To encrypt the plaintext $m \in \mathbf{Z}_n$ a random $r \in \mathbf{Z}_{n^2}$ is generated and m is then encrypted as

$$c = g^m \cdot r^n \bmod n^2.$$

While the decryption of a $c \in \mathbf{Z}_{n^2}$ works as follows

$$m = \frac{(c^\lambda \bmod n^2) + 1}{n} \cdot \lambda^{-1} \bmod n.$$

The Qredo Network makes extensive use of the Paillier Cryptosystem as it utilized in both its MPC Protocol and the Threshold Decryption scheme which are employed by the MPC Nodes as described in Section 4.3 and Section 4.4.

3.3 | Post-Quantum Cryptography

The security of almost all public-key cryptosystems in use today relies on computational assumptions such as the Integer Factorization (IF) and Discrete Logarithm (DL) problems as the foundation of their security. These are problems that today's classical computers cannot solve. In 1994, Shor[24] showed that both IF and DL problems are easy to solve on a quantum computer, based on the laws of quantum physics. As a consequence, almost all currently deployed public-key cryptosystems will become completely insecure if quantum computers become a practical reality.

According to NIST in its Report on Post-Quantum Cryptography[11]: "It will take significant effort to ensure a smooth and secure migration from the current widely used cryptosystems to their quantum computing resistant counterparts. Therefore, regardless of whether we can estimate the exact time of the arrival of the quantum computing era, we must begin now to prepare our information security systems to be able to resist quantum computing."

Most experts have a range of quantum computing strong enough to crack today's cryptosystems being on the horizon anywhere from five to twenty years. It should also be stated that quantum computation only speeds up a brute-force key search by a factor of a square root, so any symmetric algorithm can be made secure against a quantum computer by doubling the key length, i.e., take AES from 128 bits to 256.

The Qredo Network uses post-quantum cryptography in its Encrypted Envelope format, as detailed in Section

[4] https://en.wikipedia.org/wiki/Paillier_cryptosystem

4.5.

3.4 | Multi-Party Computation (MPC)

Multi-party computation (MPC) is a branch of cryptography which deals with scenarios of multiple distrustful parties performing a single computation. There is a vast amount of recent research into applying MPC techniques to digital signing, with immediate applications to securing crypto assets.

MPC can be used to provide a threshold signature functionality in the following way:

1. Several parties follow a specific protocol to generate multiple independent secrets, which are never shared.

2. These secrets are used in another protocol to produce a public key, and if the protocol continues, a single digital signature.

The simplest, yet arguably most useful application of MPC signing is 2-of-2 threshold scheme from[16], where a single wallet address containing crypto assets is controlled by two secrets, both of which are required to produce a signature. On a first attempt to instantiate such a scheme one might envision splitting a private key into parts (using Shamir Secret Sharing Scheme[23], for instance) and recombining them on each signing attempt. An important distinction of MPC signing, however, is that private key is never instantiated explicitly. By not ever generating the private key into one whole form under the control of one actor, the security of systems exercising custodial responsibility over crypto assets can increase by orders of magnitude.

In short, the defining feature of MPC signing is that the private key never has to be reconstructed in order to generate a public key, or a digital signature. The secrets held by a participant in the MPC protocol that are necessary to run the MPC protocol are much less sensitive than a raw private key in a sense that they are not, taken has a whole, self-sufficient to reproduce a signature which enables spending or moving crypto assets. In the case these secrets are compromised by an attacker, these client secrets are of no use as an attacker is unable to produce a valid signature with them.

4 | CRYPTOGRAPHIC PROTOCOLS OVERVIEW

4.1 | BLS Signatures

BLS Signatures[7] are a short signature scheme based on the computational Diffie-Hellman assumption on certain elliptic and hyper-elliptic curves ("BLS Signatures"). The BLS signature (Boneh-Lynn-Shacham) scheme makes use of PBC to generate very short signatures that can be just the x coordinate of a point $P \in_1$. Working in an elliptic curve group provides some defense against index calculus attacks (with the caveat that such attacks are still possible in the target group G_T of the pairing), allowing shorter signatures than other systems for similar levels of security. BLS signatures have become the subject of much work as they are seen as a possible way forward to solve privacy issues within cryptocurrencies through a process of signature aggregation. The Qredo Network makes extensive use of BLS Signatures. The Qredo Network implements from[3] a multi-signature with public-key aggregation scheme overlay with an M-of-N Threshold Scheme as the basis for collecting accountable signatures from Custodians which demonstrate an granting of approval over the transfer, swap or loan of a crypto asset.

In a simple example, given a secret key sk, a public key $pk = g^{Sk}$, a message m, a hashing-into-the-curve function H, and a bilinear pairing e:

- Key Generation: sk is a random integer over the field, $pk = g^{Sk}$

- Signature: $S = H(m)^{sk}$
- Verify: $e(H(m), pk) = e(S, g)$

Biliniarity is evident as the signature:

$$e(H(m), pk) = e\left(H(m), g^{sk}\right) = e(H(m), g)^{sk} == e\left(H(m)^{sk}, g\right) = e(S, g)$$

but is also unique and deterministic, something missing from ECDSA.

4.1.1 | BLS MSP Scheme

In June of 2018 Dan Boneh, Manu Drijvers and Gregory Neven published[3] that constructs a pairing-based multi-signature scheme with public-key aggregation $\mathcal{MSP}$ based on BLS signatures with formal security proof. The scheme is secure in the public key model and assumes the hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_2$ and $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$. This scheme is used as the foundation method of aggregation of signatures and public keys resident within messages on the Qredo Network blockchain.

4.2 | Supersingular Isogeny Key Encapsulation

Supersingular Isogeny Diffie–Hellman key exchange (SIDH) and the key encapsulation protocol, SIKE[5] (derived from SIDH)[19], are post-quantum cryptographic algorithms used to establish a secret key between two parties over an otherwise insecure communications channel. SIKE boasts one of the smallest key sizes of all post-quantum key encapsulations; with compression, SIKE uses 2688-bit public keys at a 128-bit quantum security level.

Since it is the only post-quantum cryptography protocol which is constructed on elliptic curves, hybrid cryptography protocols can be derived from SIKE and classical elliptic curve cryptography (ECC) to make the transition towards post-quantum cryptography more convenient and practical.

SIKE is in the round two finalist category of the NIST Post Quantum Cryptography Standardization competition and the cryptographic community is confident that it will be standardized.

A key encapsulation mechanism (KEM) is a set of three algorithms:

- key generation (KeyGen)
- encapsulation (Encaps)
- decapsulation (Decaps)

and a defined key space, where

- KeyGen(): returns a public and a secret key (pk, sk)
- Encaps(pk): takes pk as input and outputs ciphertext c and a key K from the key space.
- Decaps(sk, c): takes sk and c as input, and returns a key K or ERROR. K is called the session key.

[5] Supersingular Isogeny Key Encapsulation - NIST Submission

SIKE uses Hofheinz transformation on SIDH to achieve CCA security. Let $p = 2^{e_A}3^{e_3} - 1$, and let E be a supersingular elliptic curve defined over a field of characteristic p. E can also be defined over $\mathbb{F}_{p^2}$ up to its isomorphism. An isogeny $\phi : E \rightarrow E'$ is a non-constant map from E to E'.

An isogeny map is defined by its degree and kernel. The degree of an isogeny is its degree as morphism. An isogeny with degree ℓ map is called ℓ-isogeny. Let G be a subgroup of points on E which contains $\ell + 1$ cyclic subgroups of order ℓ. This subgroup is the torsion group $E[\ell]$ and each element of this group is corresponding to an isogeny of degree ℓ; accordingly, an isogeny also can be identified by G, i.e., the kernel of isogeny.

This section provides a brief presentation of the SIKE protocol. We refer readers to [12] and [13] for more detailed explanation of the supersingular isogeny problem and the base key-exchange protocol which the SIKE is constructed upon.

The Qredo Network implements SIKE for key encapsulation utilized through the Encrypted Envelope format (covered in Section 4.5).

4.3 | Multi-Party Computation Protocol with Trustless Setup

In 2018 Rosario Gennaro and Steven Goldfeder published[16] which outlined a threshold signature scheme to enable distributed digital signature generation among n players such that any subgroup of size $t + 1$ can sign, whereas any group with t or few players cannot. Many cryptographic frameworks for MPC leave the setup of the key material to a trusted dealer, an external party assumed to be trusted by all players. Relying on a trusted dealer is not an ideal situation to instantiate either an MPC protocol or a Threshold Decryption scheme. Simply stated, the trusted dealer could be compromised from the outset, and the entire paradigm is then compromised. Qredo Network implements 'trustless' schemes for both its Multi-Party Computation and Threshold Decryption schemes.

In regards to digital assets, a multi-party computation (MPC) protocol based on a threshold signature scheme enables n parties to share the power to issue digital signatures under a single public key. Generally, the goal is to produce signatures that are compatible with an existing centralized signature scheme. In a threshold scheme the key generation and signature algorithm are replaced by a communication protocol between the parties, but the verification algorithm remains identical to the verification of a signature issued by a centralized party. The Qredo Network utilizes this multi-party computation (MPC) protocol to produce signatures through the interaction of the MPC Nodes that are responsible for the running the multi-party computation protocol specifically. The advantage of this instantiation of MPC is that it is 'trustless', meaning no centralized dealer is required to setup the protocol, and hence create a security vulnerability. The MPC protocol outlined in[16] makes use of the Paillier Cryptosystem. It allows n players to generate an ECDSA public key PK and secret shares w_i for said key in a trustless setup.

Any group of t players can pool their resources together to issue an ECDSA digital signature the verifies under PK, without revealing any information about the shares w_i. For the trustless setup, each player generates a (t, n) SSS $(j, x_{i,j})$ for a random value u_i and the shares are distributed among the players. Each player combines the received shares, obtaining a share w_i for a (t, n) SSS of $x = \sum_{i=1}^{n} u_i$. The public key $PK = x.G$ is easily computed by broadcasting the values $u_i.G$ and combining them. The SSS shares w_i can easily be converted to additive shares using the appropriate Lagrange coefficients when t players decide to issue a signature. The signature scheme for a message with hash m can be split into two logical blocks, tied to the two components of an ECDSA signature $S = (r, s)$. The first part aims to agree on a $R = k^{-1}.G$, where $k = \sum_{i=i}^{t} k_i$ for random $k_i \in \mathbf{Z}_q$ chosen by each player. This is achieved by masking k with a random $\gamma = \sum_{i=i}^{t} \gamma_i$, as $k\gamma = \sum_{i,j=1}^{t} k_i\gamma_j$. The terms where $i = j$ can be easily computed by each player. As for the mixed products $k_i\gamma_j$, they are converted to a sum of two shares $\alpha_{i,j}, \beta_{i,j}$ using the Multiplicative to Additive (MtA) share conversion protocol detailed in the next section. This conversion allows each player to combine

Alice	Bob
$c1 = E_A(a)$	
$c1 \rightarrow$	
	Generates random $m \in \mathbf{Z}_q$
	$c2 = c1 \otimes b \oplus m = E_A(ab + m)$
	$\beta = -m \bmod q$
	$\leftarrow c2$
$\alpha = D_A(c2) = ab + m$	

TABLE 1 MtA

its $k_i\gamma_i$ term and the $\alpha_{i,j}, \beta_{j,i}$ terms to an additive share of $k\gamma$ which can be broadcast without leaking any information about the k_i and γ_i. Now, the players share $\gamma_i.G$ and combine these values to compute $\gamma.G$. The final value for R can be computed as $(k\gamma)^{-1}.(\gamma.G)' = k^{-1}.G$. The component r for the signature is simply $R_x \bmod q$.

The aim of the second part is to agree on $s = k(m + rx) = km + r(kx)$. Note that m and r are known to each player and k can be broken down into its shares k_i, allowing each player to compute a share for km simply as $k_i m$. Agreeing on a kx is slightly more challenging, but it can be achieved using the same technique used above to compute $k\gamma$. It is enough to use the shares for x instead of the ones for γ. Using this procedure each player can compute an additive share for kx and add it to the already computed $k_i m$ to compute its complete signature share. The final value of s can thus be computed as a sum of the players shares, completing the signature $S = (r, s)$. If the signature is valid for the public key agreed above the protocols terminates successfully.

4.3.1 | Multiplicative to Additive (MtA)

The Multiplicative to Additive share conversion protocol converts two multiplicative shares $a, b \in \mathbf{Z}_q$ to two additive shares $\alpha, \beta \in \mathbf{Z}_q$ s.t. $ab = \alpha + \beta$. The scheme uses any homomorphic encryption scheme, such as the Paillier cryptosystem, as its base building block.

Let Alice and Bob be the actors holding, respectively, a and b. Let E_A be the Paillier encryption using Alice public key. Alice initiates the protocol by encrypting her share as $c_A = E_A(a)$ and sending it to Bob. Bob chooses a random $\beta \in \mathbf{Z}_n$, computes $c_B = c_A^b \cdot E_A(-\beta) = E_A(ab - \beta)$ and sends c_B back to Alice. In the final step Alice decrypts c_B and sets $\alpha = D_A(c_B)$ (see Table 1).

It's worth pointing out that $\alpha + \beta = ab - \beta + \beta = ab \bmod n$. The equality holds If a and b are small enough for ab to be less than n.

4.4 | Threshold Decryption with Trustless Setup

Threshold decryption in a public key cryptosystem with n parties means a minimal number of parties is required to decrypt a ciphertext and excludes the situation where a single party (holding the decryption key) is able to decrypt all sensitive information.

The Paillier Cryptosystem can be adapted to achieve (t, n) threshold decryption, where each of the n players receives a share of a private key, so that t players can pool their resources to decrypt a ciphertext without leaking any information about the shares. For the sake of simplicity this section presents the scheme with a trusted dealer setup. The next section will introduce a 'trustless' setup, which removes the risk of a compromised trusted dealer. The trusted dealer generates a random $\beta \in \mathbf{Z}_n$ and a (t, n) Shamir Secret Sharing of $\lambda\beta \in \mathbf{Z}_{n^2}$, adding $\theta = \lambda\beta \bmod n$ to the public key. To decrypt a ciphertext c, the players compute the Lagrange coefficients to convert their SSS shares to an additive (t, t) secret sharing. Instead of applying the coefficients directly to the shares s_i, they apply them to the c_i^s, computing $c^{\lambda\beta}$. The plaintext is computed as in the scheme presented above, with the caveat of substituting c^λ with $c^{\lambda\beta}$ and λ^{-1} with θ^{-1}

The Threshold Decryption scheme can be modified to have a trustless setup. This section only contains a brief overview of the steps. For the full details we refer to [25].
The first step in this setup is to generate the biprime n for the factorization trapdoor. The players generate SSS for random numbers p_i and q_i and each player combines the received shares, computing a SSS for the full p and q. The shares for these two SSS can be multiplied together to compute a $2t$ SSS of n, which can be safely revealed so n can be tested for biprimality. The second step is to compute the shares for an SSS of λ from the shares for p and q and to generate an SSS for a random β in the same way the SSS for p or q were generated. Each player multiplies their shares to compute a $2t$ share of the private key $\lambda\beta$ and then reduces this share modulo n to compute a $2t$ share of the public key. Finally, the public key share are broadcast and the public key is reconstructed.

The Qredo Network utilizes this threshold decryption scheme to secure the MPC Client Node or MPC Server Node secrets mentioned in Section 2. The scheme requires least two sets of the total deployed class of MPC Nodes (Client or Server) and those nodes must be from the same class. The scheme secures the MPC Node's sets of secrets necessary (secured across a shared storage layer) to run the MPC protocol to generate public keys or signatures over wallets containing crypto assets using the threshold decryption scheme from Section 4.4. Each MPC Node in the subsets are physically isolated from the other and communicate over private links. When they become instantiated for the first time collectively on the network, each deployed MPC Node of the same class works collectively to create a shared public key (for encryption) with a private key (for decryption) that can only be created with the cooperation of a threshold of other MPC Nodes in the subset (i.e., at least one other node of its class - Client or Server). This cooperation is only obtained for legitimate requests from Qredo Core Clients that have the Transaction Right as expressed by ownership of the crypto asset, definitively stated on the Qredo Network blockchain.

In simple terms, multiple MPC Nodes need to validate that the request to invoke the MPC protocol comes from the legitimate owner of the crypto asset before any one node can decrypt their respective MPC protocol secrets and primitives necessary for that node to run the MPC protocol. They each do this independently, verifying the state of ownership as declared on-chain, separately interrogating the blockchain as an oracle of recorded ownership of crypto assets. The security consequence being that multiple MPC Nodes of the same class need to be collectively compromised in order to reveal any cryptographic primitives that enable the invocation of the MPC protocol. The individual secrets for each MPC Node necessary to run the threshold decryption scheme are stored in the MPC Node's FIPS 140-2/3 HSM, which are a component of the secured appliance that each MPC Node runs on. Each MPC Node is physically isolated from other MPC Nodes; the MPC Nodes are located in different Tier 4 data centers

who have achieved SOC 3 compliance.

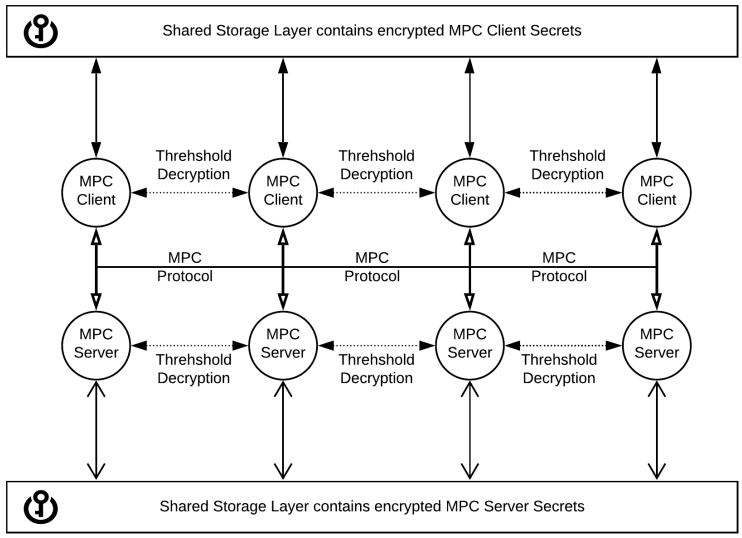

FIGURE 3 MPC Nodes

4.5 | Encrypted Envelope

At its core, taking a cue from RFC 5652 which describes the Cryptographic Message Syntax[6], the Encrypted Envelope format is a cut-down version of the above that is designed to deliver the following features:

- Authentication (using encryption key encapsulation)
- Message integrity
- Non-repudiation of origin (using digital signatures)
- Privacy and data security (using AES-256 encryption)

The Encrypted Envelope format utilizes the Protocol Buffer (protobuf) format[7] to serialize the structured data messages that move between Qredo Core Clients in order to validate the identity of a trusted counterparty.

Starting with a plaintext, an AES 256 bit key is randomly generated and used to encrypt the plaintext to ciphertext. For each recipient of the message, the AES key is encapsulated using the recipient's SIKE public key located in their ID Document, which can be found by querying the blockchain using the recipient's Account Code identifier.

Once the AES key is encapsulated, once for each recipient, the message plus encapsulated encryption keys are concatenated together and serialized using protobufs. The resulting binary data is digital signed by the originator of the message using their BLS secret key whose corresponding public key is written into their ID Document. The detached digital signature is concatenated to the existing binary object and the serialized a final time (using protobufs) for transmission to the intended recipients.

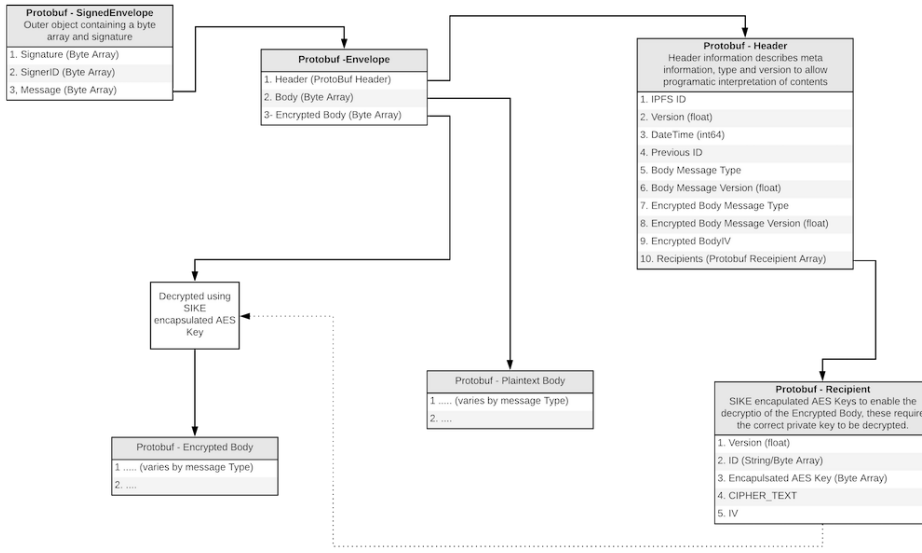

FIGURE 4 Encrypted Envelope Format

[6] https://tools.ietf.org/html/rfc5652

[7] Protocol Buffer: https://developers.google.com/protocol-buffers/

5 | OPERATIONS

5.1 | Initialization and Group Setup

On the Qredo Network, crypto assets are organized by Funds. This distinction lives inside the organizational framework and logic of the Qredo Core Clients, but does not exist within its blockchain, which only tracks crypto assets and validates statements on chain, and the corresponding digital signatures over those statements, made by actors on the system known only by their Account Codes.

Each Qredo Core Client, upon initialization, creates an ID Document, as covered in 2.1, which contains unique BLS and SIKE public keys. The public keys that come from the generated public/private key pairs themselves are generated from a seed value. This enables each actor on the Qredo Network to have their own Account Code, which the SHA-256 hash on the ID Document.

Actors on the Qredo Network can organise themselves into Groups. These groups are represented by an ID document containing the Account Codes of the actors in the group along with their BLS digital signatures that correspond to the BLS public keys found within each actor's individual ID Document. The signatures are necessary to note active participation in the Group.

There are two main groups types: Custodian Groups and Principal Groups. Custodian Groups are a set of individual actors who are prepared to act as Custodians over crypto assets within a Fund on the Qredo Network as described in Section 2.2. A Custodian Group ID Document will also contain a threshold statement in addition to Account Codes and signatures of its members. The threshold statement is simply the number of digital signatures required out of the total in the Custodian Group for an action on crypto assets within a Fund to which the Custodian Group is assigned to be recognized as being legitimate by other clients or nodes within the network (ex: 2 out 3).

Principal Groups can also be instantiated which describe which other actors on the network who can take Principal responsibilities over a Fund that enables them to invoke a transfer, atomic swap, or perform some form of collateral pledge or sweep. Other inanimate items can be grouped as well, such as the white list of wallet addresses that a Fund can settle to. All of these groups have their own ID Documents, and have rules governing how these documents are updated.

5.2 | Qredo Network blockchain

The Qredo Network blockchain does not have a native token. The Qredo Network blockchain serves as a consensus driven immutable record of asset ownership.

In a simple example illustrated below in Figure 5, Principals operating Qredo Core Clients employ the BLS public key aggregation scheme to declare a desired, open, future state in regards to a crypto asset under their control. As laid out in the example from Section 2, a transfer of a crypto asset requires the process to be invoked by the Principal, agreed to by the Beneficiary and approved by the Custodians.

The Principals create aggregated public keys and include them in digitally signed messages created by them declaring the desired future state. Custodians and Beneficiaries complete and finalize the state by digitally signing those same messages created and signed by the Principal initially, in reference to a particular crypto asset. The signatures from all participants in the transaction are aggregated into a few single digital signatures. If the state was completed, the aggregated digital signatures will verify with the aggregated public keys created by the Principal as a part of the Principal's opening messages.

From Figure 5 below, the following will outline a simple example whereby a crypto asset resides in a Fund with a

Custodian Group which contains two Custodians, and the thresholds of approvals to transfer the crypto asset require that both Custodians agree to the action (2 out of 2 agree). The first step requires the Principal to create a message on chain which declares the following: What specific Custodians, and what threshold of approvals from those Custodians is required in order to complete what action (transfer, atomic swap, collateralize, etc.) regarding a specific crypto asset.

Starting with Message 1 from Figure 5 below, the digitally signed message contains the Asset ID, a list of Custodians, what action they must approve (Transfer), the threshold policy (2 out of 2) and a single aggregated public key labelled Aggregated Public Key 1 in the diagram below. Aggregated Public Key 1 consists of the aggregation of the Principal's public key with an aggregated public key that consists of the two Custodians listed. This action declares to the other nodes on the network exactly which Custodians and how many of them in the group need to digitally sign this message in following messages after the Asset Transfer message is created by the Principal at a later date.

Once the Custodian Setup Message is accepted and confirmed into the blockchain, the Asset ID's corresponding public key created by the MPC Nodes can be used to form a wallet address, and the Principal is enabled to fund the wallet address with crypto assets.

The next process begins when the wallet address is funded, and the Principal wishes to transfer the crypto asset to another counterparty on the Qredo Network. Message 2 is created and digitally signed by the Principal. Message 2 contains an Asset ID, and the Account Code of the Beneficiary of the transfer, and an aggregated public key labelled Aggregated Public Key 2 in the diagram below. Aggregated Public Key 2 consists of the Principal's public key aggregated together with the intended Beneficiary's public key.

The next step falls to the intended Beneficiary who is alerted to the transfer through the Matrix communication protocol. The Beneficiary can inspect the confirmed Asset Transfer Message on chain. Assuming it is acceptable, the Beneficiary duplicates the message and creates its own digital signature over the message contents. It uses its signature over the message and aggregates its signature together with the Principal's signature in the preceding message to create an Aggregated Signature 2. This Aggregated Signature 2 can be verified with Aggregated Public Key 2 as shown in the diagram below. Nodes on the network interrogating the blockchain will confirm this transfer of the crypto asset was accepted by the Beneficiary. However, the nodes on the network will not deem the transfer complete until the Custodians approve it. These are the Custodians referred to in Message 1 described previously.

The Custodians receive requests to approve the transfer over the Matrix communication protocol which includes Proof of Coin. Interrogating the blockchain, they can ascertain which crypto asset is being transferred, and to which Beneficiary. Assuming they approve of the transfer, the Custodians copy the plaintext of Message 1, sign the message and submit it to the network. In Figure 5 below, these are represented as Message 4 and Message 5. With these messages confirmed on the blockchain, other nodes on the network will deem the transfer complete. With that, the Beneficiary named in the transfer becomes the new owner (Principal) of the crypto asset. Depending on the policy over the Fund to which the crypto asset is placed, this enables the new Principal to create a signed settlement message on chain for the MPC Nodes, who, assuming Settlement Custodians approve and the recipient wallet address is whitelisted, will create the necessary transaction on the underlying blockchain. Or, the new Principal is free to custody the crypto asset within the Qredo Network, trade, collateralize, or atomically swap the asset when the opportunity arises.

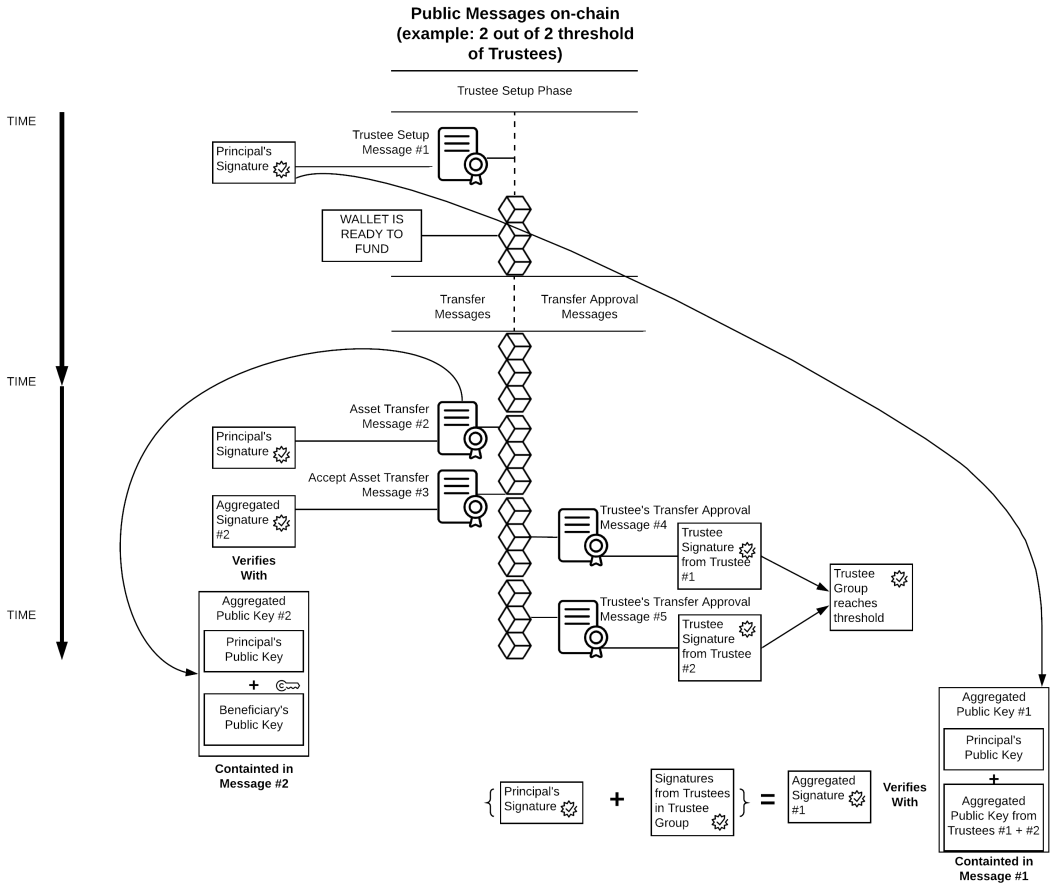

FIGURE 5 Qredo Network blockchain

5.2.1 | In Practice

In practice, the above example can be condensed into a just a few messages with aggregated public keys and signature. Messages 2,3,4 and 5 can all be concatenated together with one aggregated public key and aggregated signature. The advantage inherent in the BLS signature scheme, signature aggregation, means that all of the information about the lifecycle of the crypto asset held by a Principal can be condensed into two 32-bytes signatures.

Much more complex conditions and financial transactions can be instantiated in a similar simple manner. As an example, a Creditor role can be assigned over a crypto asset, which bypasses the need for Custodians to approve a transfer, and the transfer can be invoked by the Creditor solely. This enables crypto assets to be synthetically held on a Principal's balance sheet, but made available to the Creditor at their discretion. Crypto assets can also have different sets of Custodians, one set for transfer and clearing across the network (Transfer), and another that must approve requests to create transactions on the underlying blockchain (Settlement).

We refer the reader to the Qredo Network documentation website at https://docs.qredo.network for detailed

information on message types, structure and sequence.

6 | CONCLUSION

Our decision to build the Qredo Network on top of the open-source freely available (and hence, auditable) components such as Apache Milagro and Tendermint reflects the reality that auditable systems open to source code and peer review have a historical track record of being much more secure than closed source counterparts.

The Qredo Network's long term goals and ambitions are to usher in a new paradigm for crypto asset custody to further the growth of decentralized industries globally. We hope that you the reader will join us in this effort.

References

[1] W. Ashford, 2015. Experian chooses uk authentication startup for gov.uk verify. https://www.computerweekly.com/news/4500260479/Experian-chooses-UK-authentication-startup-for-GovUK-Verify.

[2] A. Boldyreva, *Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme*, Public Key Cryptography, 2003.

[3] D. Boneh, M. Drijvers, and G. Neven, 2018. Compact multi-signatures for smaller blockchains. https://eprint.iacr.org/2018/483.

[4] D. Boneh and M. Franklin, *Identity-based encryption from the weil pairing*, Springer-Verlag, 2001, pp. 213–229.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps*, IACR Cryptology ePrint Archive **2002** (2002), 175.

[6] D. Boneh, S. Gorbunov, H. Wee, and Z. Zhang, *BLS Signature Scheme*, Internet-draft draft-boneh-bls-signature-00, Internet Engineering Task Force, Feb. 2019, Work in Progress.

[7] D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the weil pairing*, J. Cryptology **17** (2004), 297–319.

[8] S. Bowe, 2017. Bls12-381: New zk-snark elliptic curve construction.

[9] S. Bowe, A. Gabizon, and I. Miers, 2017. Scalable multi-party computation for zk-snark parameters in the random beacon model. https://eprint.iacr.org/2017/1050.

[10] V. Buterin, *Exploring Elliptic Curve Pairings*, Medium (2017), https://medium.com/@VitalikButerin/exploring-elliptic-curve-pairings-c73c1864e627.

[11] L. Chen, S. Jordan, D. Moody, Y.K. Liu, R. Peralta, R. Perlner, and D. Smith-Tone., Report on post-quantum cryptography. https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf.

[12] C. Costello, P. Longa, M. Naehrig, J. Renes, and F. Virdia, 2019. Improved classical cryptanalysis of the computational supersingular isogeny problem. https://eprint.iacr.org/2019/298.

[13] L.D. Feo, D. Jao, and J. Plût, 2011. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. https://eprint.iacr.org/2011/506.

[14] D. Freeman, M. Scott, and E. Teske, *A taxonomy of pairing friendly elliptic curves*, J. Cryptography **23** (2010), 224–280.

[15] S. Galbraith, K. Paterson, and N. Smart, *Pairings for cryptographers*, Discr. Appl. Math. **156** (2008), 3113–3121.

[16] R. Gennaro and S. Goldfeder, *Fast multiparty threshold ecdsa with fast trustless setup*, Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, New York, NY, USA, 2018, CCS '18 pp. 1179–1194.

[17] R. Gennaro and S. Goldfeder, 2019. Fast multiparty threshold ecdsa with fast trustless setup. https://eprint.iacr.org/2019/114.

[18] A. Joux, *A one round protocol for tripartite diffie–hellman*, J. Cryptology **17** (2004), 263–276.

[19] P. Longa, 2018. A note on post-quantum authenticated key exchange from supersingular isogenies. https://eprint.iacr.org/2018/267.

[20] M. Scott, 2002. Authenticated id-based key exchange and remote log-in with simple token and pin number. https://eprint.iacr.org/2002/164.

[21] M. Scott, 2016. A novel multi-factor id-based designated verifier signature scheme. https://eprint.iacr.org/2016/1151.

[22] M. Scott and B. Spector, 2015. The carnac protocol – or how to read the contents of a sealed envelope. https://eprint.iacr.org/2015/576.

[23] A. Shamir, *How to share a secret*, Commun. ACM **22** (Nov. 1979), 612–613.

[24] P. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, Proc. 35th Ann. IEEE Symp. Foundations Comput. Sci. (1994), 124–134.

[25] T. Veugen, T. Attema, and G. Spini, 2019. An implementation of the paillier crypto system with threshold decryption without a trusted dealer. https://eprint.iacr.org/2019/1136.