VaVite: Verifiable Information Exchange for Virtual Asset Service Providers

Vytautas Tumas, Robert Norvill, Damien Magoni, Radu State

VaVite: Verifiable Information Exchange for Virtual Asset Service Providers

Vytautas Tumas
SEDAN - SnT
University of Luxembourg
Luxembourg, Luxembourg
vytautas.tumas@uni.lu

Robert Norvill
SEDAN - SnT
University of Luxembourg
Luxembourg, Luxembourg
robert.norvill@uni.lu

Damien Magoni
LaBRI - CNRS
University of Bordeaux
Talence, France
magoni@labri.fr

Radu State
SEDAN - SnT
University of Luxembourg
Luxembourg, Luxembourg
radu.state@uni.lu

***Abstract*—Virtual Asset Service Providers (VASPs) have recently been faced with increased compliance costs as both national and international bodies bring their due diligence compliance requirements inline with those of traditional financial entities. The 'travel rule', as defined by the Financial Action Task Force (FATF) is prominent amongst these and has created a need for efficient compliance solutions for VASPs. In this paper we improve upon current travel rule compliance solutions by utilising a mixed centralised and decentralised approach to provide trust and reduce the compliance burden on VASPs. Moreover, we provide a generic system capable of ensuring compliance with FATF recommendations as well other sanctions and embargoes that impact VASP trading relationships.**

***Index Terms*—Blockchain, FATF, Identity Verification, Travel Rule, VASP.**

I. Introduction

International regulatory efforts and guidelines have been increasing in recent years as governments and international organisations make changes to account for the rise of cryptocurrencies and Virtual Assets (VAs). Notable changes include the 5th Money Laundering Directive in the EU [1], and updates to the Financial Action Task Force (FATF) recommendations, which place a regulatory burden on Virtual Asset Service Providers (VASPs) which previously only applied to banks [2]. According to FATF, a VASP is any natural or legal person which transfers, holds, or otherwise handles virtual assets on behalf of another (natural or legal person). Moreover, they define a VA as "any digital representations of value that can be digitally traded or transferred and can be used for payment or investment purposes" [2]. In practice, VAs include cryptocurrencies and digital representations of equities and financial instruments, and VASPs are the entities that handle these, such as cryptocurrency exchanges. The above definitions bring a large number of businesses within the scope of the recommendations and as such, there is a need to find efficient and secure solutions for compliance.

As FATF recommendations are not legally binding, there are likely to be differences in how they are implemented in law in different countries. Such recommendations are implemented as part of a jurisdiction's Know Your Customer (KYC) laws. KYC laws require financial institutions to acquire their customer's identity and actively monitor for illicit financial activities. FATF recommendations and recent EU directives are increasing the KYC compliance burden for VASPs. Cross-jurisdiction differences are made more likely by various embargoes and financial sanctions are taken into account, which can have changing and serious impacts on business operations [3] [4]. The problem for VASPs is compounded when we consider that they are likely to make use of payment routing protocols, which offer rapid, low-cost cross-border transfers. The paths they use to transfer value are often over multiple hops using a variety of currencies, both fiat and virtual, with nodes located in different jurisdictions. In order to remain compliant in today's global environment, VASPs need to be able to know the identity of those they do business with, and be able to control the entities and jurisdictions they engage with.

In this paper we aim to solve some of the compliance problems faced by VASPs through a blockchain-based system of identity verification and path compliance enforcement.

- We provide the means to comply with FATF recommendations as well as other international rules and regulations. The proposed solution avoids centralised points of failure and eases the KYC burden for VASPs.
- We allow VASPs to specify a set of rules that they will operate under. Rules are defined as a set of constraints, where each constraint identifies some property of a VASP, for example the country in which the VASP resides . The rules are verified and stored on-chain, making them tamper-proof.
- We provide assurance and non-repudiation for all participants of each payment path through the use of signed on-chain attestations.

The rest of this paper is organised as follows: Section II takes a more in-depth look at the travel rule compliance solutions. Section II-D introduces the relevant state of the art proposals for FATF travel rule compliance and discusses state of the art academic works in the area of identity verification. Section III details the protocol and blockchain system. Section IV provides a theoretical network topology and performance analysis. Finally, we discuss further research topics and conclude our work in section V.

II. Related work

A. Ethereum

Existing approaches and the solution proposed in this paper rely on an Ethereum blockchain for various parts of their functionality. Ethereum as a system was launched in 2016 and is the vehicle for one of the world's most popular cryptocurrencies: Ether. Ethereum uses immutability and system-wide consensus to ensure that all stakeholders can trust what is recorded on-chain, with dishonesty being visible to participants as well as being financially disincentivised [5].

Ethereum provides the functionality to operate a permissioned blockchain, which works much like the public main-chain, with the difference those who can participate and add blocks to the chain is curated.

However, its major contribution to the blockchain world was the implementation, and subsequent popularisation of smart contracts. As programs stored on the blockchain, the execution of smart contracts is carried out by all participants with the results of execution being agreed upon by consensus. As such, the results of running smart contract code can be trusted.

Smart contract code is arbitrary, and as such there are numerous use cases which include; creation of one's own Ethereum-based cryptocurrency (token), access control, and identity verification. The latter of which is key for the topic of this paper [5].

B. OpenVASP

The OpenVASP protocol heavily leverages Ethereum. It proposes to use a standardized Ethereum smart contract, referred to as the VASP contract, for the identification of a VASP. The address of the contract deployed by the VASP serves as a unique identifier of a VASP. Two approaches to trust are used in OpenVASP: direct and indirect. Direct trust forms the primary trust model, whenever two VASPs establish their business relationship, they provide first-hand evidence of their identity, which is used to perform customer due diligence. This step does not have to be made public, however a large number of trusted peers demonstrates the VASP's good standing. The indirect approach to trust requires a trusted third party, such as recognized self-regulatory organisations or trade associations, attest to the identity and the licensing of a VASP. The identity claims, either by trusted VASPs or credible third parties are signed and stored in the contract of a VASP and can be revoked by the issuer at any time.

Using the VASP's public key poses new challenges in case the corresponding private key is compromised. Smart contracts deployed on Ethereum blockchain are permanent, thus in case the private key is compromised, there is no way to remove the contract. Thus, potentially, there will be a live contract on the blockchain, which all the participants on the network will have to know not to interact with. Although not explicitly discussed in the whitepaper, in lieu of any other solution it seems the VASP would have to inform each of its peers about a compromised private key, that, in turn, will have to revoke all identity claims. Furthermore, if the VASP is to continue doing business, it will have to deploy a new VASP contract on the blockchain, changing the unique VASP ID. In a recent proposal [6], the VASP contract was extended with ownership transfer functionality. This allows for the owner of the contract to be updated if necessary. However, anyone aware of the owner's private key can act as the owner of the contract, thus allowing a malicious entity to transfer the ownership to itself.

C. TRISA

While OpenVASP aims to provide compliance with the FATF recommendations, the system relies on traditional, potentially manual, identity verification methods, with the whitepaper stating that for identity verification "first-hand evidence is available..." [7]. Exchange of evidence in this manner is non-standardised and may be time consuming and costly.

TRISA also utilises CAs to perform KYC checks. A CA acts as an authorized third party which verifies the identity of a VASP. In a standard known as Extended Validation Know Your VASP (EV KYV), the CA performs due diligence checks on the VASP in their jurisdiction, performs fraud and sanction checks, validates the VASP's business and performs active monitoring, auditing and reporting of the VASP. Two VASPs operating in different jurisdictions can establish a trusted relationship by verifying the certificates of each other with the corresponding CA. EV KYV eases the KYC burden on individual VASPs by using the CA to provide due diligence. However, it requires that the CA is always available, making it a centralised point of failure. In case the CA is made unresponsive legitimate VASPs will not be able to have their identity verified or ensure that certificates issued by the CA have not been revoked through the Online Certification Status Protocol (OCSP) [8] or the Certificate Rejection List (CRL) [9].

OCSP requires the receiver of a certificate to check with the issuer of the certificate whether the certificate has been revoked. The protocol has been further extended to include stapling, which requires the certificate owner to make periodic OCSP requests to the CA to acquire a time-stamped signature in order to prove the validity of the certificate. On the other hand, a CRL is a periodically published list of rejected certificates by the CA. To ensure a certificate has not been revoked, the receiver of the certificate, downloads the rejection list from the issuer of the certificate, and ensures the certificate is not present in the list. Both approaches rely on the CA to maintain knowledge of refuted certificates. In addition to providing a centralised point of failure, the burden of maintaining extra infrastructure could potentially deter institutions from taking on the role of a CA.

D. Identity and Identity Management

The X.509 standard [9] defines the format of public key certificates. X.509 is used for a range of applications, from securing network connections via TLS/SSL [10] to digital signatures. The certificate contains information identifying the owner and a signature of a CA that has verified the

content. The blockchain technology provides new methods for certificate management, which offer alternatives to OCSP and CRL, we will discuss these efforts bellow.

Wang *et al.* [11] propose a method of storing X.509 certificates on a blockchain, with multiple CAs signing each certificate and certificates only being accepted if the subject of the certificate also signs it. This method provides a heightened security of standard SSL certificate issuance as the theft of the private key of a single CA will not result in the acceptance of fraudulent certificates. However, multiple CAs signing each certificate is likely to significantly increase the overhead of the system, especially if each CA is required to carry out full identity verification for the subject. Moreover, certain transactions are required to be signed by all CAs in the system, which presents a potential problem if one of the CAs is offline or unreachable.

Chen *et al.* present a similar system called certChain [12], they cite security weaknesses stemming from, amongst other things, centralisation. The system stores X.509 certificates on-chain and defines entities called 'bookkeepers' who act as miners/verifiers for the blockchain. Bookkeepers pair with CAs and they have a shared reputation within the system. However, the exact nature of a bookkeeper and its relationship with the CA is unclear. While CAs are financially incentivised it is not clear whether bookkeepers are considered to share the same profit motive.

Kubilay *et al.* [13] propose a solution called certLedger which aims to reduce the required trust in CAs by moving certificate and revocation lists onto the blockchain. The authors cite a number of attacks against centralised CAs as motivation. certLedger relies on a board of organisations to verify a CA within the system. Board member public keys are hard coded into the smart contract, meaning the contract and its previous state contain the list of trusted CAs needed to be redeployed. This is likely to be a cumbersome task as in effect the entire program needs to be edited and re-added to the blockchain. Retaining the original state may also be complex, depending on the number of records stored.

In order for VASPs to fulfill the FATF recommendations and the Travel Rule in particular, Hardjono *et al.* [14], [15] define an architecture similar to the one defined by the Security Assertion Markup Language (SAML), where the VASP is the relying party and a blockchain record is used for identification. The attribute provider, holding sources of truth about the subject, is called the Claims Provider (CP). A CP delivers signed claims to VASPs regarding the relevant subject, thereby relieving the VASP from having to deal with data and algorithms. In addition, they propose a consortium arrangement for VASPs to establish a Claims Exchange Network (CEN), in which VASPs can deliver signed claims (obtained from their CPs) and public-key information or certificates to other VASPs in a secure and confidential way.

III. VaVite Protocol

We propose a new system, which combines strengths of OpenVASP and TRISA to provide irrefutable identity verification, simple KYC, and verifiable compliance for FATF recommendation as well as various national and international country and currency restrictions. Our solution provides increased trust while avoiding the need for high availability of CAs.

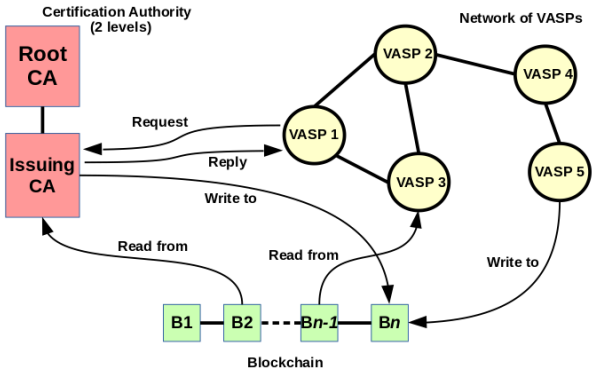

Fig. 1: VaVite high-level architecture.

We outline the high level architecture in Figure 1. The system is composed of three components: *Certificate Authority, Ethereum Blockchain and a VASP Network*. A VASP wanting to join the network issues a Certificate Signing Request to the CA. The CA, verifies the identity, issues a certificate and places it on the blockchain through its smart contract. VASP can then establish business relationships with other VASPs, by reading their identity information from the blockchain. VASP peerings establish a transitive network, akin to Lightning [16] or Raiden [17] Networks, that enables VASPs to establish multi-hop payment paths. Once a path is established it is written to the blockchain by the VASP from which the payment originates. In the remainder of this section we will discuss in detail on how the system provides irrefutable guarantees on identity, conformation to rules and participation in payments.

A. VASP Identity

VASP identity is represented by a X.509 certificate issued by a CA and stored on the blockchain (discussed in more detail in the next sub-section). Similarly to OpenVASP, the Etheruem address of the certificate is derived from the public key of the VASP. In case the public/private key pair is changed, the address will also change. This would require VASPs to re-establish their peerings. We address this by introducing a static identifier for each VASP called VASP Identity Code (VIC).

Inspired by Business Identity Codes (BIC) [18] a VIC takes the following form: 4 letter institution code, 2 letter country code, 2 letter location code (city code), 3 letter branch code (optional). It can include digits for some exceptional cases. VICs are thus 8 or 11 letter long. Table I compares the BIC of Luxembourg's national bank and the VIC for the hypothetical VASP1 entity, which also operates out of Luxembourg, in Luxembourg city.

The mapping $VIC \rightarrow EthAddr$ is stored, by the CA, in a smart contract, the address of which is known to all

TABLE I: Comparison of the Business Identity Code *vs* the VASP Identity Code.

BIC:	BCEE	LU	LL
VIC:	VASP	LU	LL

TABLE II: JWT Token fields.

JWT Code	Name	Value Description
sub	Subject	VIC of the VASP creating the token
aud	Audience	VIC of the intended recipient of the token
exp	Expires At	Time when the token expires
iat	Issued At	Time when the token was issued
nbf	Not Before	Time from which the to ken is valid
jti	JWT ID	Unique token identifier

participants in the network. The VIC can then be used to exchange identities with other VASPs.

When two VASPs A and B exchange their VICs neither of them have a guarantee that the VIC actually belongs to the other VASP. To ensure VIC authenticity we adopt JSON Web Token (JWT) [19] standard. The standard defines a method for tamper-proofing information for exchange between two parties, that is provided by digital signature using a secret token or a public/private key pair. A token signed with a private key allows anyone with the corresponding public key to verify the ownership of the private key. In Table II we outline the standard fields of the token and the corresponding values. As the token is issued by the VASP to prove the identity of the VASP we have excluded the Issuer (*iss*) field.

In Algorithm 1 we outline the process of validating VASP JWT token. Upon receiving a JWT token, VASP uses the Subject field of the token to look up the certificate on the blockchain. An absence of a certificate indicates that the token is not valid. The VASP uses the public key from the certificate to verify the ownership of the private key. Finally, if the VASP is the intended audience and the token is not expired, the token is deemed valid.

Algorithm 1 JWT Token Validation

procedure TOKENVALID($\mathbb{T}$)
  $cert_{peer} \leftarrow$ ETHEREUM.GETCERTIFICATE($\mathbb{T}.sub$)
  if ABSENT($cert_{peer}$)**or!**SIGVALID($\mathbb{T}, cert_{peer}$) **then**
    return false
  end if
  vic_{vasp} ▷ VIC of the VASP validating the JWT token
  if $\mathbb{T}.aud \neq vic_{vasp}$ **then**
    return false ▷ Audience VIC does not match VASP VIC
  end if
  $curTime$ ▷ Current time
  return $\mathbb{T}.nbf > curTime < \mathbb{T}.exp$
end procedure

With the knowledge of VASP identity we can further discuss how the identity is verified.

B. Identity Verification

The protocol uses a permissioned blockchain as a decentralized public key infrastructure, by storing proof of identity in the form of X.509 certificates on-chain. In keeping with standard SSL certificate practice, identity certificates must be signed, in this context by a the financial regulatory body for the jurisdiction in which the entity is operating. For example, a VASP operating in the USA should have its identity verified the Securities and Exchange commission (SEC), who would issue a certificate that acts as proof of identity, it is stored on the blockchain. Once a valid certificate is stored on-chain and associated with a VASP, the VASP can participate in the network. In order not to place too great of a burden on regulators we allow for certificate chains. A third party with a certificate issued by the regulator is authorised to carry out identity verification.

To maintain a high level of trust in our system, certificate chains are restricted to a length of two. CAs are granted verifier certificates by the jurisdiction's regulatory body, which can be used to verify identities, however CAs are not allowed to issue verifier certificates themselves. While the certificates of CAs and VASPs are stored on-chain the regulator's root certificate is publicly available through some other channel, likely their own server. This gives the regulator the flexibility to participate in the chain or not, as is appropriate for their role.

Algorithm 2 VASP registration with a Certificate Authority

procedure REGISTERVASP($\mathbb{C}, \mathbb{R}$)
  if not ETHEREUM.ADDRESSFREE($\mathbb{C}.addr$) **then**
    return false
  else if not (IDVALID($\mathbb{C}$) $\wedge$ RULESVALID($\mathbb{C}, \mathbb{R}$)) **then**
    return false
  end if
  $cert \leftarrow$ SIGNCERTIFICATE($\mathbb{C}$)
  ETHEREUM.STORECERTIFICATE($\mathbb{C}.addr, cert$)
  ETHEREUM.STOREADDRMAPPING($\mathbb{C}.addr, cert.VIC$)
  ETHEREUM.STORERULES($\mathbb{C}.addr, \mathbb{R}$)
end procedure

Smart contracts are used to store a list of verified entities who are allowed to participate in the system. Each CA operates a smart contract for its jurisdiction, which holds the certificates and CRL for VASPs within that jurisdiction. If an entity's certificate cannot be found in the smart contract for the jurisdiction in which it operates, it is considered invalid, and should not be interacted with. As such, in order to join the VaVite network (Algorithm 2) a candidate VASP has to create a Certificate Signing Request (CSR) and prepare a set of rules (covered in subsection III-C). The CSR contains necessary information to carry out KYC checks in the requester's jurisdiction. In addition, the CSR contains an Ethereum address, derived from the corresponding to the public/private key pair they generate for use in the system, the process is detailed by Wood [5]. The address acts as a unique identifier of the entity on the network, and will be used to look up the identity of

the owner of the address. Finally, the candidate also includes their VIC in the CSR, which acts as a permanent identifier for the VASP.

Upon receiving the information, the CA ensures there is no address clash and verifies the identity of the VASP. The process of verifying the identity of a VASP will depend on the laws in the jurisdiction of the VASP. The CA then signs the certificate and stores it on the blockchain, the certificate is associated with the Ethereum address and the VIC of the VASP. The VIC is used to retrieve VASPs information as, unlike the certificate and address, it never changes. X.509 certificates specify a period of validity, this is set by the issuer to coincide with the date when the requesting entity's identity must be re-checked in accordance with due diligence laws, thus ensuring all entities are verifiable at all times. Finally, the $VIC \rightarrow EthAddr$ mapping and $EthAddr \rightarrow cert$ mappings are saved.

We assume that the CSR can uniquely identify a VASP and it cannot be forged. In case the public/private key pair has been compromised or the certificate needs to be replaced the VASP can request that the CA updates their certificate. The VASP has to send an identity proving CSR to the CA, which, in turn, carries out necessary identity checks and issues a new certificate. If the previous certificate was not expired the CA revokes the old certificate by updating the CRL held in the smart contract. The CA then adds the new certificate to the chain. The smart contract associates the certificate with the VASPs VIC and Ethereum address by the smart contract by updating the $VIC \rightarrow EthAddr$ and $EthAddr \rightarrow cert$ mappings with the new address. If the CSR is for a standard certificate renewal no revocation is necessary and the $VIC \rightarrow EthAddr$ mapping does not need to be updated.

Revocation also takes place when a VASP leaves the system. The CA accepts revocation requests of from either the VASP itself or a recognised regulatory body for the jurisdiction in question. The CA will remove the mapping to the certificate, thus suspending VASP's ability to participate in the system.

The methods detailed above rely on the honest behaviour of CAs'. Under our system CAs are incentivised to behave honestly through financial means. CAs are licensed by their respective national regulators, and paid for their identity verification work by VASPs wishing to be issued or re-issued a certificate. Any dishonest behaviour by a CA will be visible to all stakeholders due to the immutable on-chain record of certificate issuances. Misbehaviour would result in fines or loss of license, with a loss of license meaning a loss of revenue as they can no longer verify VASPs. VASPs themselves are similarly incentivised to behave honestly. The system reduces their compliance costs and allows for faster, trusted interactions between one another. The paths they have been part of, and their rules at any given point are all recorded on-chain. A deviation from the rules would result in fines or expulsion from the system, which would negatively impact costs and revenue, especially if it is assumed that VASPs may choose to only do business with other VaVite participants.

Algorithm 3 Peering with another VASP

```
1: procedure PeerVASP(id_peer)
2:     if TokenValid(id_peer) ∧ RulesMatch(id_vasp, id_peer) then
3:         SavePeering(id_peer)
4:         ethereum.SubscribeToRules(id_candidate)
5:     end if
6: end procedure
```

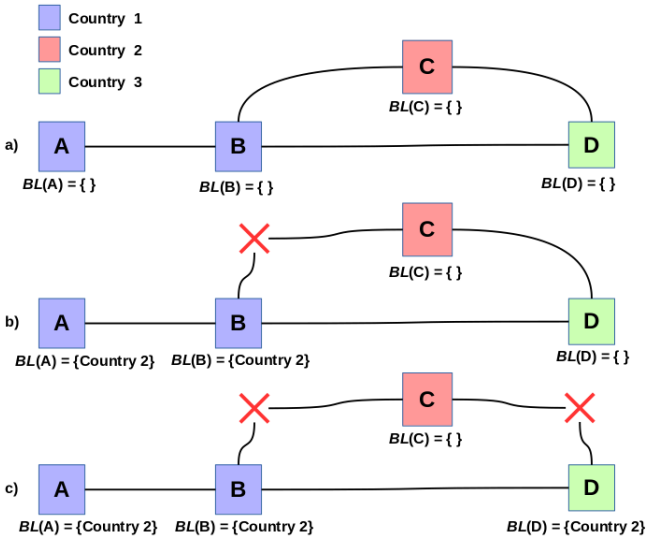

Fig. 2: Blacklist rules.

C. Rule Based Path Verification

Imagine a VASP network outlined in Figure 2, VASPs A and B in jurisdiction of Country 1, VASP C in Country 2 and VASP D in Country 3. Due to increase in terrorism in Country 2 to prevent the funding of those activities Country 1 has decided to restrict the trade of virtual assets with Country 2, furthermore it announces that it will seize participating in virtual asset trade with any country with relationship to Country 2. VASPs A and B add Country 1 to their blacklist, VASP D, to keep its business relationship with B also blacklists Country 2. However, B does not have any assurance that D blacklisted Country 2. We propose a rule based system to address this challenge. Inspired by firewewalls and MPLS, each VASP has a publicly accessible list of rules they adhere to when conducting business relationships with other VASPs. The rules identify assets and countries a VASP is unwilling to conduct business with. They apply not only to its immediate peers, but also any other VASP through which a payment might travel. In our example, VASPs in country 1 blacklisting country 2, VASP D responds by updating its rules to match those of VASP B. As the rules are publicly accessible VASP B can be sure that D updated their rules. In the remainder of the subsection we will discuss how conformation to rules is guaranteed, and participation in business relationships is made irrefutable.

As part of the registration process with the CA (Algo-

Algorithm 4 Initiation of payment path establishment

```
1: procedure EstablishPath(vic_dst)
2:     vic_vasp                ▷ VIC of the VASP initiating the path
3:     vic_signed ← NewJWT(vic_dst)
4:     ℙ ← {VICS : [vic_signed], SIGS : []}
5:     SendPacket(ℙ, vic_dst)
6: end procedure
```

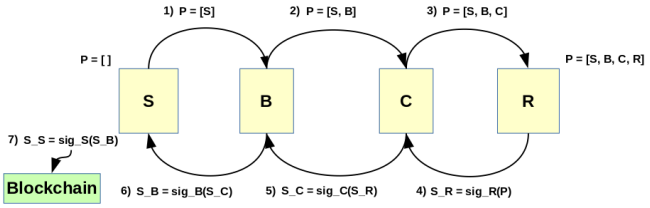

Fig. 3: Path establishment.

rithm 2) VASP submits a set of rules that indicate high-risk assets and countries the VASP is unwilling to conduct business with. These restrictions are dictated by laws and agreements in the VASP's jurisdiction. The CA's role is to ensure there are legitimate grounds for the constraints and the VASP is not abusing its position. Upon validation the CA places the rules on the blockchain via a well defined Smart Contract that it owns. The rules can then be retrieved using VASPs Ethereum address.

To establish a business relationship with another VASP (Algorithm 3), the VASP reads the candidate's rules from the blockchain and ensures they match their own rules. If the rules match and a relationship can be established, the VASP subscribes to the events on the blockchain related to the rules of the candidate. As a result, it will be notified of any changes in the rules of the peer, and will be able to act accordingly.

Path establishment is the initialisation and reservation of a network path between a source and a destination prior to any data transmission. It requires the set-up of forwarding information in every node traversed by the path. Packets then all follow the same path until the connection is closed. In our solution, path establishment is required for regulatory reasons, to ensure that every node in a given path is identified and trusted. Path establishment requires to send a specific setup message which will travel from the source to the destination and vice-versa while installing stateful information in every crossed intermediate node.

It is imperative to ensure that all VASPs in a multi-hop payment path adhere to each other's rules. A VASP has first-hand evidence of its peers rules, but not of VASPs beyond. For example, in Figure 3, VASP S has no information about VASPs beyond VASP A, thus it has not assurance of rule adherence. We address this by introducing an extra layer between payment path establishment and payment execution: payment path verification. The algorithm allows to securely collect the identities of participants in the path, and verify the rules of each.

The originator VASP S creates a Path Verification Packet $\mathbb{P}$ (Algorithm 4), with its VIC in the participant list (Step 1 in Figure 3), and sends it to the recipient R using the underlying routing algorithm of the network.

Each VASP upon receiving this packet (Steps 1-3) appends their VIC to the list of participants and forwards it to the next VASP. Once the packet arrives at the destination, the receiver validates the rules and signs the packet. The signature serves two purposes, it indicates the packet is no longer modifiable and that the signer has verified the rules of each participant, found no clashes and agrees to participate in the payments on the path. VASP R then forwards the packet back to the originator, each node verifying and signing the packet along the way (Steps 4-6). When the packet reaches VASP S, it verifies that all participants have signed the packet and saves it on the blockchain (Step 7). Thus creating irrefutable agreement between all VASPs to participate in a transaction over the verified path.

Algorithm 5 Handling a path establishment packet

```
1: procedure HandlePacket(ℙ)
2:     if len(ℙ.SIGS) = 0 then
3:         vic_vasp  ▷ VIC of the VASP handling the packet
4:         vic_signed ← NewJWT(vic_vasp)
5:         ℙ ← AppendVIC(vic_signed)
6:     else if len(ℙ.SIGS) > 0 ∧ RulesValid(ℙ) then
7:         sig ← NewJWT(ℙ)
8:         ℙ ← AppendSIG(sig)
9:     else
10:        return                ▷ Invalid rules; Drop packet
11:    end if

12:    if DestinationReached(ℙ) then
13:        if len(ℙ.VICS) == len(ℙ.SIGS) then    ▷ All
   participants have signed the packet
14:            ethereum.PersistPaymentPath(ℙ)
15:            return
16:        else if RulesValid(ℙ) then
17:            sig ← NewJWT(ℙ)
18:            ℙ ← AppendSIG(sig)
19:        else
20:            return            ▷ Invalid rules; Drop packet
21:        end if
22:    end if
23:    ForwardPacket(ℙ)
24: end procedure
```

In the following section we will discuss how the system impacts VASP network formation and explore theoretical overheads of path establishment.

IV. Theoretical Analysis

Rule based path verification will have an impact on the network topology and introduce overheads for payment processing. In this section we will provide a theoretical analysis

of these effects, we leave the experimental analysis for future work.

We assume all VASPs in the network are properly incentivised to behave honestly, they follow the rules that have been verified by the CA. VASPs peering only with other VASPs whose rules match their own will produce a graph with multiple connected components, each of which will have identical rules. The intuition behind this is as follows, assume peered VASPs A and B with corresponding rule sets $\mathbb{R}_A$ and $\mathbb{R}_B$. From Algorithm 3 we know that $\mathbb{R}_A \equiv \mathbb{R}_B$. A new VASP C joins the network and establishes a business relationship with B, which implies $\mathbb{R}_B \equiv \mathbb{R}_C$. From transitive relations we can say that $(\mathbb{R}_A \equiv \mathbb{R}_B \wedge \mathbb{R}_B \equiv \mathbb{R}_C) \implies \mathbb{R}_A \equiv \mathbb{R}_C$.

When VASP changes its rules, as seen in Figure 2, its peers have to update their rules to match or terminate their relationship, disconnecting from the connected component. However, the choice whether to disconnect or update can be dictated by multiple factors. For example, if VASP C blacklisted B, VASP D might be reluctant to follow suit, as B allows for further interaction with A. On the other hand, if VASP C provides more trade opportunities than B and A combined, it might be more valuable to terminate the relationship with B.

The total time required to establish a path depends on the number n of VASP nodes on the path (ranging from 2 to a given maximum m). In figure 3, $n = 4$. The message going to the destination will require $n-1$ hops. If we define t_{delay} as the average delay between two nodes then:

$$t_{forward} = (n-1)t_{delay} \tag{1}$$

On the return path, every node will have to verify all other VASP identities thus requiring $n-1$ local read operations from the blockchain followed by a token check which is a signature verification. The source node will then have to write the path to the blockchain for bookkeeping. If we define t_{read} the average time for reading a record on the blockchain, t_{sign} the average time for signing a message, t_{check} the average time for verifying a signed token, t_{rules} the average time to check the rules, and t_{write} the average time for writing a record to the blockchain then:

$$\begin{aligned} t_{return} &= n(n-1)(t_{read} + t_{check}) \\ &\quad + n(t_{rules} + t_{sign}) + (n-1)t_{delay} + t_{write} \end{aligned} \tag{2}$$

Thus the total time for path establishment is given by:

$$\begin{aligned} t_{total} &= 2(n-1)t_{delay} + n(t_{rules} + t_{sign}) \\ &\quad + n(n-1)(t_{read} + t_{check}) + t_{write} \end{aligned} \tag{3}$$

Also note that t_{write} depends on the size of the blockchain network, as a write operation requires its broadcast over the whole network to reach a consensus.

In an Internet map collected by Hoerdt *et al.*, 90% of the links between two routers or one router and one host have a delay below 40ms [20]. The average number of hops between two hosts in this map is 11. Thus, we can use

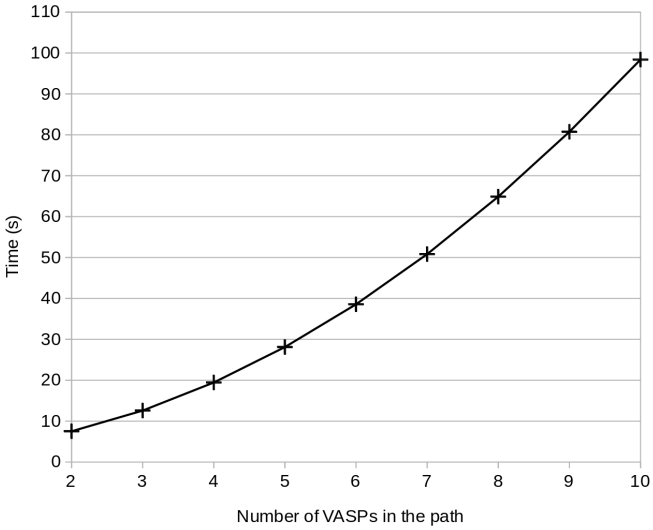

Fig. 4: Path establishment time *vs* number of VASPs in path.

440ms as a reasonable average value for t_{delay}. Jansma *et al.* have compared in [21] the performances of ECC and RSA for signing and verifying data and have found for a 100KB file, 590ms/860ms for 283-bit ECC keys and 210ms/10ms for 3072-bit RSA keys. The processor used was an Intel P4 at 2GHz which is of the same order of magnitude than the performances of nowadays server processors typically running at 3GHz. Thus we can use 590ms as a realistic value for t_{sign} and 860ms for t_{check} if we assume that the signed data is not bigger than 100KB and the processor's frequency is above 2GHz. Checking the rules requires comparing each element of the path to each element of the blacklist. This time t_{rules} can be considered negligible compared to the others. Recently, Norvill *et al.* have experimented a permissioned blockchain deployed internationally for implementing a KYC data sharing platform [22]. They have found that the average time for a local READ operation takes 39ms while a consensus WRITE operation takes 3659ms. As their system has a similar functionality to ours, we will therefore use 39ms for t_{read} and 3659ms for t_{write} as first indicators in our numerical application. Given the equation 3 and the above values, we have plotted the average time of path establishment *vs* the number of intermediary VASPs ranging from 2 (i.e., only one originator VASP and one beneficiary VASP) to 10. The results are shown in figure 4, and exhibit a quadratic time increase. This is due to the fact that the quadratic term n^2 of the equation is tied to t_{check} which is numerically significant. A typical path with 2 intermediaries will take 19.4s to complete under the above assumptions.

V. Conclusion & Future Work

In this paper we propose a new system, which combines the strengths of OpenVASP and TRISA to provide irrefutable identity verification, simple KYC, and verifiable compliance for FATF recommendation as well as various national and international country and currency restrictions for VASPs. Our

solution provides increased trust while avoiding the need for high availability of CAs. Trust and availability is ensured through the use of a private blockchain.

Compliance is assured through a rule-based system which allows VASPs to define a set of blacklist rules which enable them to remain compliant with all the laws and international stipulations that are relevant to them. By ensuring they are only involved in payment paths were all participants are in accordance with each others' rules.

Future work includes formally defining rules and the inclusion of 'soft' rules which would allow path participants to trade with entities disallowed by other path participants while keeping the restriction that disallowed entities cannot be in the path, and extending identity verification model to include VASP end-clients.

Acknowledgements

The authors acknowledge the financial support from Ripple UBRI [23].

References

[1] E. Union, "Directive (eu) 2018/843 of the european parliament and of the council," *Official Journal of the European Union 156*, pp. 43–74, 2018.

[2] FATF, "International standards on combating money laundering and the financing of terrorism & proliferation," 2012. [Online]. Available: https://www.fatf-gafi.org/media/fatf/documents/recommendations/pdfs/FATFRecommendations2012.pdf

[3] M. Frank, "Tougher u.s. sanctions make cuba ever more difficult for western firms," in *World News*. Reuters, 2019.

[4] O. of Public Affairs, "Treasury issues changes to strengthen cuba sanctions rules," September 2019. [Online]. Available: https://www.treasury.gov/resource-center/sanctions/Programs/Documents/cuba_fact_sheet_20190906.pdf

[5] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[6] D. Riegelnig, "Vasp identity," May 2020. [Online]. Available: https://github.com/OpenVASP/ovips/blob/master/ovip-0003.md#transferownerrole-1

[7] D. Riegelnig and B. Suisse, "OpenVASP : An Open Protocol to Implement FATF's Travel Rule for Virtual Assets," pp. 1–36, 2019.

[8] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and D. C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," RFC 6960, Jun. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc6960.txt

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280 (Proposed Standard), Internet Engineering Task Force, May 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[10] E. Rescorla, "The transport layer security (tls) protocol version 1.3," August 2018. [Online]. Available: https://tools.ietf.org/html/rfc8446

[11] Z. Wang, J. Lin, Q. Cai, Q. Wang, D. Zha, and J. Jing, "Blockchain-based certificate transparency and revocation transparency," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[12] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, and R. Du, "Certchain: Public and efficient certificate audit based on blockchain for tls connections," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2060–2068.

[13] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "Certledger: A new pki model with certificate transparency based on blockchain," *Computers & Security*, vol. 85, pp. 333–352, 2019.

[14] T. Hardjono, A. Lipton, and A. Pentland, "Towards a public key management framework for virtual assets and virtual asset service providers," 2019. [Online]. Available: https://arxiv.org/pdf/1909.08607.pdf

[15] ——, "Privacy-preserving claims exchange networks for virtual asset service providers," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2020, pp. 1–8.

[16] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[17] "Raiden network," https://docs.raiden.network/.

[18] I. S. 8, *ISO 9362:2014 Banking — Banking telecommunication messages — Business identifier code (BIC)*. Geneva, Switzerland: International Organization for Standardization, 2014. [Online]. Available: https://www.iso.org/standard/60390.html

[19] M. Jones, J. Bradeley, and N. Sakimura, "Json web token (jwt)," May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7519

[20] M. Hoerdt and D. Magoni, "Cartographie distribuée du cœur de l'internet," *Annales Des Télécommunications*, vol. 60, no. 5, pp. 558–587, 2005.

[21] N. Jansma and B. Arrendondo, "Performance comparison of elliptic curve and rsa digital signatures," University of Michigan, Tech. Rep., 2004.

[22] R. Norvill, C. Cassanges, W. Shbair, J. Hilger, A. Cullen, and R. State, "A security and privacy focused kyc data sharing platform," in *2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure (to appear)*, 2020.

[23] "University blockchain research initiative." [Online]. Available: https://ubri.ripple.com/