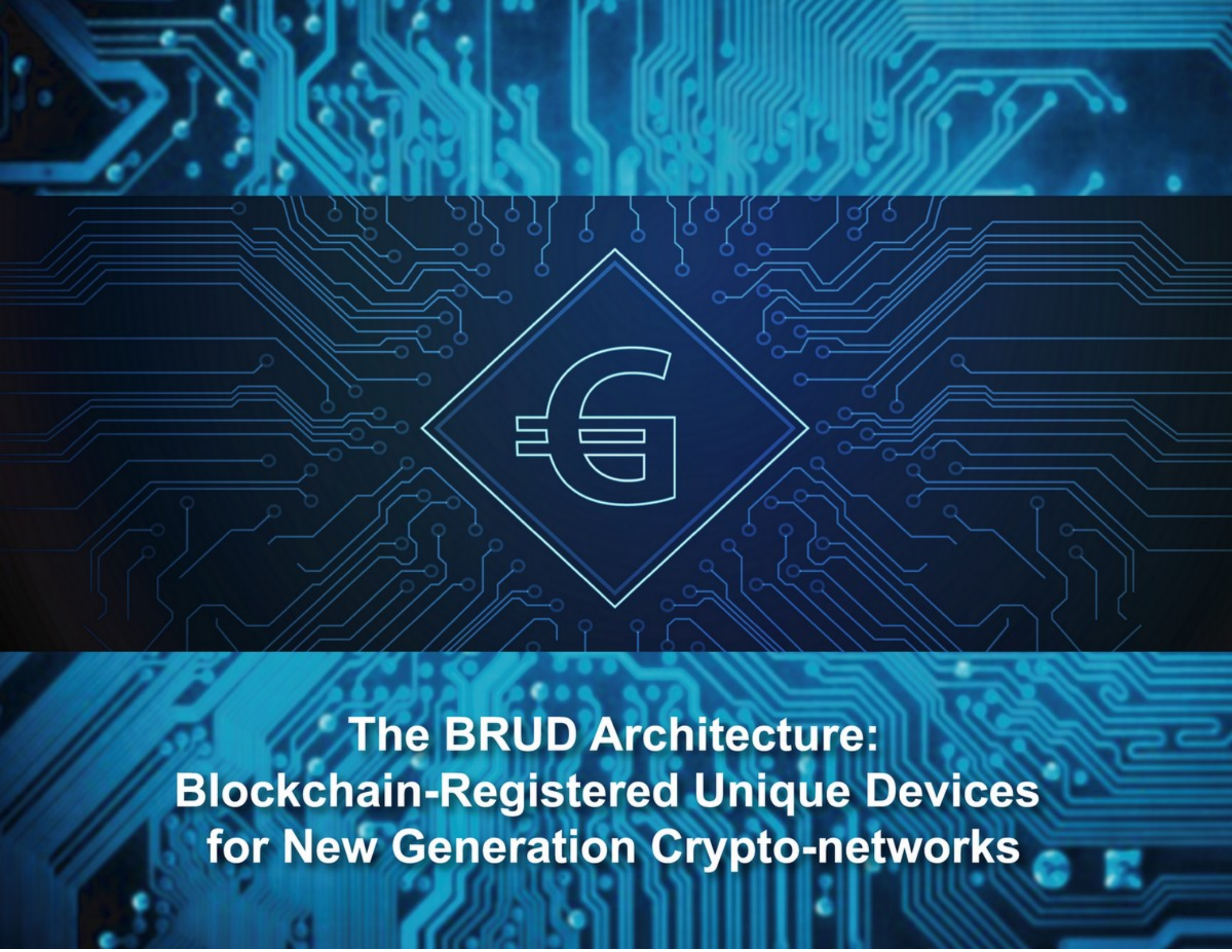


The BRUD Architecture: Blockchain-Registered Unique Devices for New Generation Crypto-networks

Table of Contents

1. Introduction....4

1.1 The Vision....4

1.2 The Underlying Technology....5

1.3 Objectives of BRUD devices....6

2. A Change in Focus in Network Security....7

2.1 Traditional Network Security....7

2.2 Identity Verification in Current Crypto-networks....8

2.2.1 Identity Verification....12

2.2.2 The Gorbyte Random Network....14

2.3 Distribution of Responsibility....16

2.4 Distributed Crypto-network Users....17

2.5 BRUD Device Uniqueness....18

2.6 Support for Virtual Private Blockchains....19

2.6.1 Subscribing to a VPB....20

3. Summary Description of a BRUD Device....21

3.1 BRUD Protocol Introduction....21

3.2 BRUD Device Registration....21

3.3 Registered Key Change....23

3.4 BRUD Device Verification at Connection Time....25

3.5 The BRUD Device Evolution....28

APPENDICES **33**
A Digital Access and Identity Classification **33**
A.1 A Progressive Scale 34
A.2 The BRUD Device in the New Crypto-network Environment 37
A.3 BRUD Device Evolution from PoDU to PoPI 38
A.3.1 The BRUD_PoDU Model 39
A.3.2 Future BRUD Device Models 40
A.3.3 The BRUD_PoPI Model 41
B Manufacturers' Opportunity **44**
C BRUD Device General Requirements **45**
D Verifications Made Possible by BRUD Devices **47**
E Hacking the System **50**
E.1 Obtaining the Manufacturer's Private Key 51
E.2 Buying Most of the BRUD Devices 51
E.3 Stealing the Private Key by Tampering 52
E.4 Stealing or Hacking most BRUD Devices 53
E.5 Denial of Service Attacks 54
E.6 Client Software Simulated 54
E.7 Client Software Hacked 55
E.8 Multiple Identity Attacks 56
Notes **57**

1. Introduction

1.1 The Vision

This document is part of the Gorbyte architecture and defines a family of software (virtual) or hardware devices, collectively and generally called **b**lockchain-**r**egistered **u**nique **d**evices (BRUD). Their function includes accessing blockchain general distributed applications (GApps), increasing the security of new generation crypto-networks, and proving the uniqueness of a node or a user, while maintaining the user's complete anonymity.

The ***BRDG*** is an instance of a BRUD device compatible with the Gorbyte architecture. The specifications of the ***BRDG*** device are proprietary.

Future BRUD models will evolve to eventually become mobile, tamper-proof, biometric, multiple-function devices.

These will provide users with the convenience of one device for accessing all distributed applications running on the **d**istributed **o**perating **e**nvironment (DOE) of new generation crypto-networks.

BRUD devices will provide the user interface, the biometric authorization, and the connection to innumerable and compatible distributed applications with the same simple interface. Examples include secret messaging, voting, financial transactions, E-commerce, physical access authorization, accessing records, proof of licenses, etc. (See section 3.5).

1.2 The Underlying Technology

New public, unpermissioned crypto-networks, such as Gorbyte, will use a cooperative consensus mechanisms. They will allow each user to become a network node and use the basic financial transaction services provided by the network for free.

Gorbyte already uses the basic concept of a Distributed Proof-of-Work to secure the network against malicious attacks. BRUD devices enhance such security by allowing the detection and prevention of majority attacks.

The additional means for preventing malicious attacks is to verify the uniqueness of each crypto-network node.

The concept of uniqueness is related to, but different from the concept of personal identity. ***Uniqueness*** can be achieved while maintaining the user's complete ***anonymity***. It is defined according to the access control and authentication classification method (introduced in Appendix A.1).

BRUD devices will implement, depending on the model, various levels of uniqueness, ranging from device, to human, to personal.

This architecture provides the flexibility for such devices to easily adapt to advances in access and authorization technology (e.g. biometric techniques).

The first model introduced by Gorbyte (BRUD PoDU model) will be a software-only virtual device and will be used to verify the uniqueness of a node/BRUD association. It will not include biometric functionality.

Future BRUD models described in this document could be USB devices, or intelligent communication (mobile) devices.

This document does not deal with security threats or defenses related to transactions and their encryption or encryption techniques.

BRUD devices cannot monitor, nor can intervene in the process of verification of transactions.

1.3 Objectives of BRUD devices

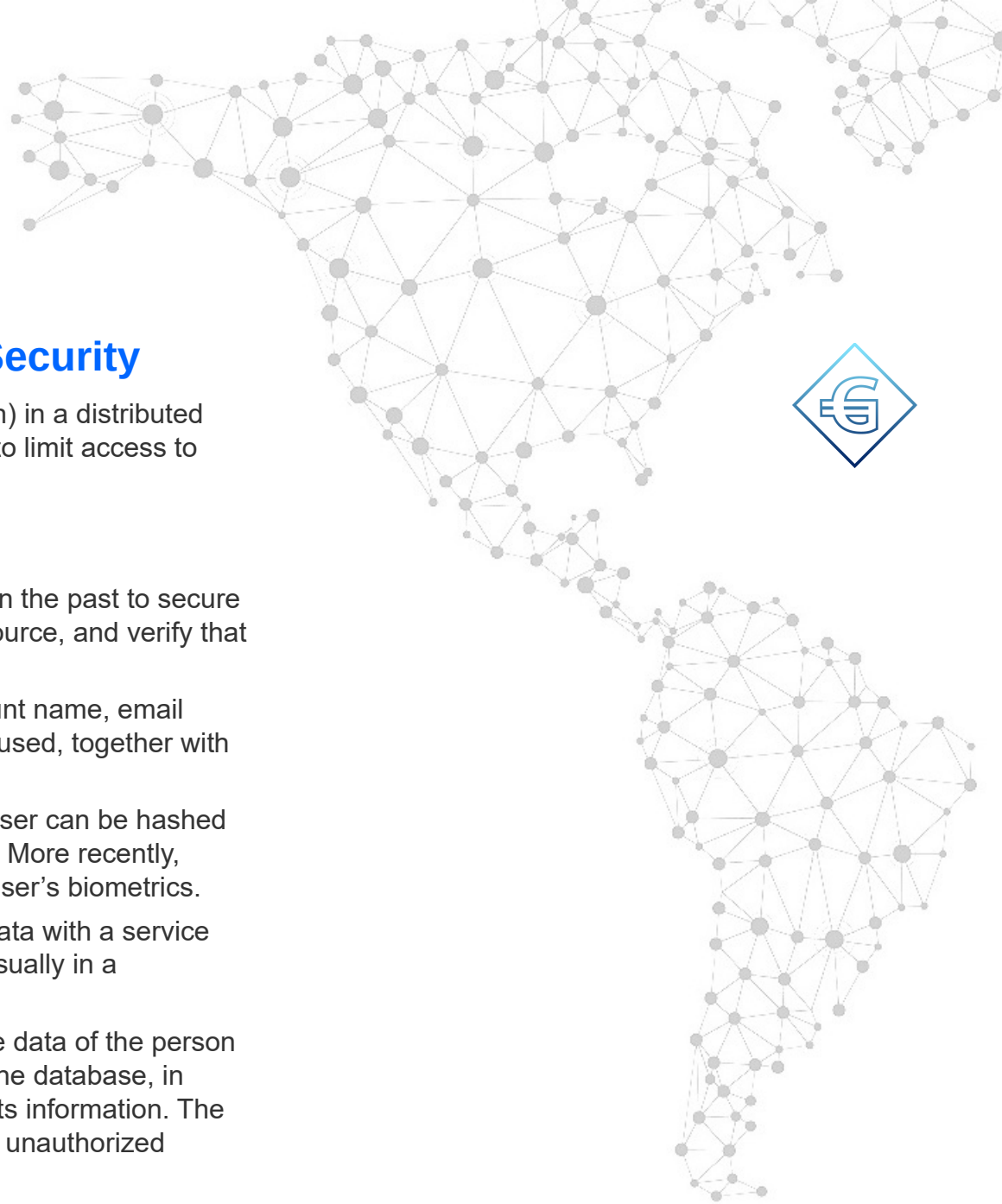
The ***main objective*** of a BRUD device is to allow secure access to the DOE environment and its innumerable general distributed applications (GApps) with a simple, common user interface (touch, voice, etc.).

A ***second objective*** is to enhance security against various forms of malicious attacks in a distributed-consensus crypto-network. BRUD device owners generally have a vested interest in the security of the network and may participate in governance decisions.

A ***third objective*** is to utilize a simple way to upgrade the access and security of BRUD devices, and a s a consequence of the network, as the related technologies improve (i.e., by model replacement).

A ***fourth objective*** is to decentralize the responsibility for guaranteeing the uniqueness of BRUD devices, by involving manufacturers and payment companies for the initial registration of BRUD devices on the blockchain[1].

In the DOE environment, when more advanced BRUD devices will be holding personal biometric information, a ***fifth objective*** is to maintain such information in a controlled, standardized and tamper-proof hardware unit, instead of a software environment with an unknown level of risk.



2. A Change in Focus in Network Security

The purpose of security (authentication and authorization) in a distributed network environment, just as in a centralized system, is to limit access to protected resources to a specified number of people.

2.1 Traditional Network Security

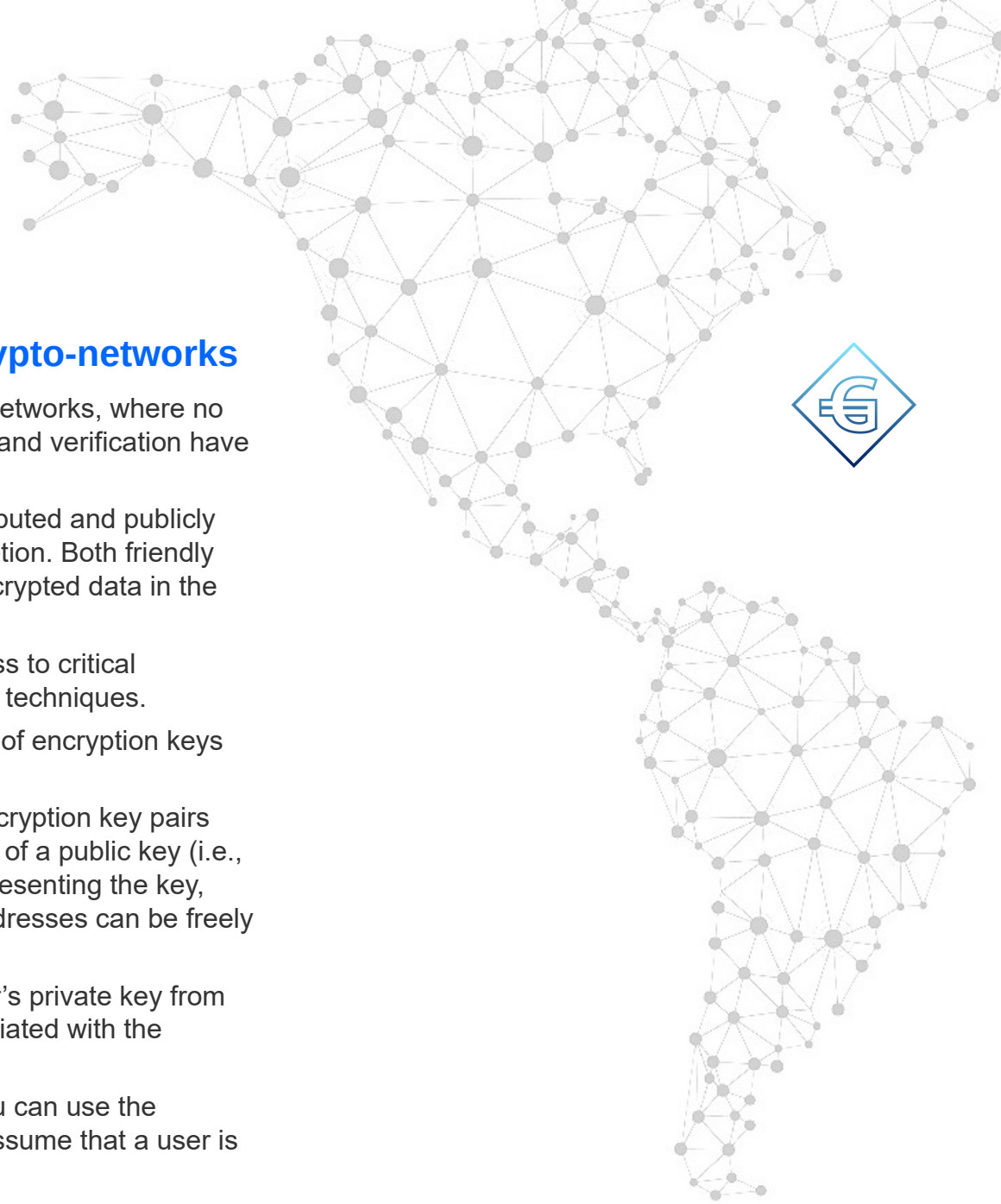
Software and hardware solutions have been developed in the past to secure resources, define the authorized group of users of a resource, and verify that a certain person is part of the authorized group.

To identify a user, some combination of password, account name, email address or biometric information have been traditionally used, together with several verification techniques.

For example, a password chosen and only known by a user can be hashed and then sent to a service provider for future verification. More recently, hardware devices have been used to collect one of the user's biometrics.

With these techniques, a person willingly shares some data with a service provider. This data is stored in a centralized database, usually in a datacenter that has added physical security.

The purpose of access control has been to verify that the data of the person requesting access or authorization matches the data in the database, in order to authorize access to a site, its resources and/or its information. The primary concern was preventing stolen identities, i.e., an unauthorized person posing as an authorized person.



2.2 Identity Verification in Current Crypto-networks

With the introduction of public, un-permissioned crypto-networks, where no central authority can exist, the requirements for security and verification have changed.

In a crypto-network, information and resources are distributed and publicly accessible, but critical information is protected by encryption. Both friendly and malicious users have access to the same raw or encrypted data in the open blockchain.

The primary concern now is allowing or preventing access to critical information, and verifying signatures, through encryption techniques.

Thus, we need to understand a little more about the use of encryption keys and crypto-network addresses.

In a crypto-network environment, users can generate encryption key pairs (public and private keys). An *address* is a standard hash of a public key (i.e., a one-way transformation of the string of characters representing the key, using one of the standard hashing techniques). Thus addresses can be freely generated.

Addresses must be secure: one cannot derive the owner's private key from the address, and one cannot derive the public key associated with the address without further data.

A user's wallet is a collection of keys and addresses (you can use the analogy of a keyring). In this context, for simplicity, we assume that a user is running a full-node.

In general, addresses are used in place of an explicit ID to maintain user anonymity.

Existing crypto-networks use addresses for several purposes, as shown in Table 1, below:

Use	Explanation
Financial transaction destination ID	Addresses are used as destination reference IDs for the two users involved in a transaction. Both the sender and the receiver must agree on the address where the money should be sent. For this function, the address should only be used once to increase security. For example, the receiver can provide a different address to each of his customers with each invoice, for them to pay the invoice amount.
Signature verification	Addresses are part of the data used to verify a signed message. From the data and the signer's address (a hash of the signer's public key), the signer's public key can be reconstructed. The signed message can then be verified
Peripheral service authentication	Addresses can be used for authentication by peripheral services (e.g., as a password required to access an online wallet).

Table 1: Uses of crypto-network addresses (continues...)

Use	Explanation
User identification	Addresses can be used as a user name in a mining pool, to receive shares of reward payments
Node/wallet ID	Addresses can be representative of (can identify) a wallet – and as a consequence a crypto-network node (i.e., the user's internet-connected device running the client software and wallet at the time). That is the user's node can be referenced by an address generated by the user. This node-address association is only temporary.

Table 1: Uses of crypto-network addresses

As we can see from the above list of uses, in current crypto-networks all a user needs for identification is an address (supported by an encryption key pair). Of course, addresses can just as easily be lost or forgotten (i.e., never come into play again).

It is up to the user to understand the different uses and to keep his/her wallet organized, so that different uses are not mixed, and so that the highest possible level of security is maintained.

Even though addresses have such an important role, one may never know who generated them, when they will be used, or if they will be used again.

An address may or may not be stored on the blockchain, depending on how it is used.

The identification function an address provides is only temporary: as long as they are needed.

Once used in a transaction, they can be seen by anyone by looking at the blockchain. For example, anyone can calculate the total unspent outputs associated to an address.

They are valuable when they have unspent outputs, they are worthless otherwise.

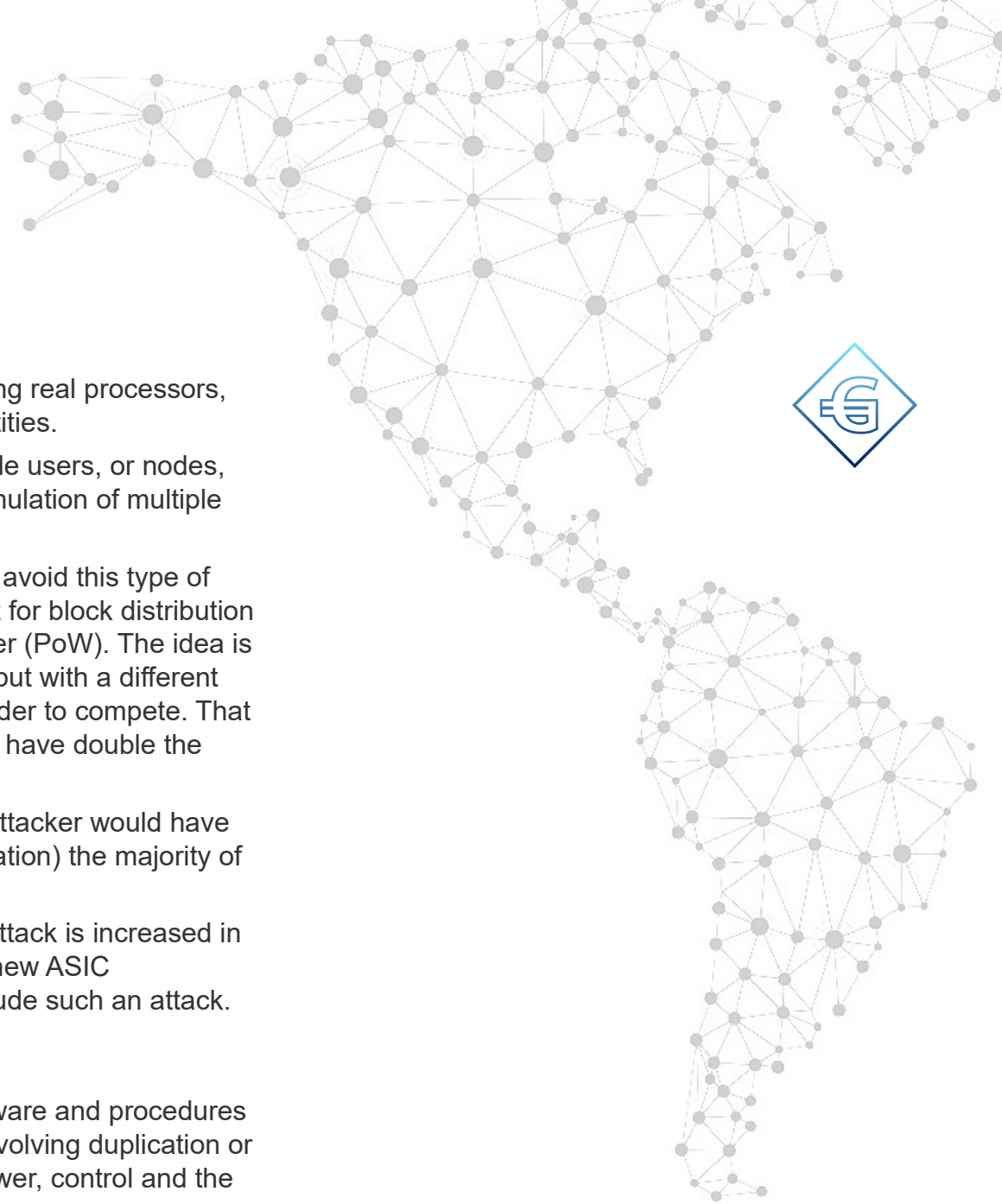
On the positive side, such level of anonymity makes it very difficult to steal someone else's identity.

Unfortunately, the ease of generating addresses and using them in transactions also provides an opportunity for foul play, as is briefly described by the following examples:

(a) Denial of service attacks

Denial of service (DoS) attacks can be mounted by generating thousands of addresses and seemingly valid transactions, without the expectation of any of them being verified and included in the blockchain. The intent in this case is to disrupt the normal functioning of the network by generating traffic load. This form of attack is particularly effective when a network, such as Bitcoin, is already backlogged.

Transaction fees, being required for all transactions, are not a deterrent to DoS attacks, since transaction fees are calculated by miners only after transactions have been broadcast, and collected.



(b) Majority attacks

Majority attacks can be mounted by taking over or hacking real processors, or by simulating nodes by generating multiple node identities.

Since multiple addresses can be used to simulate multiple users, or nodes, the ability to generate addresses at will allows for the simulation of multiple node identities (multiple identity attack).

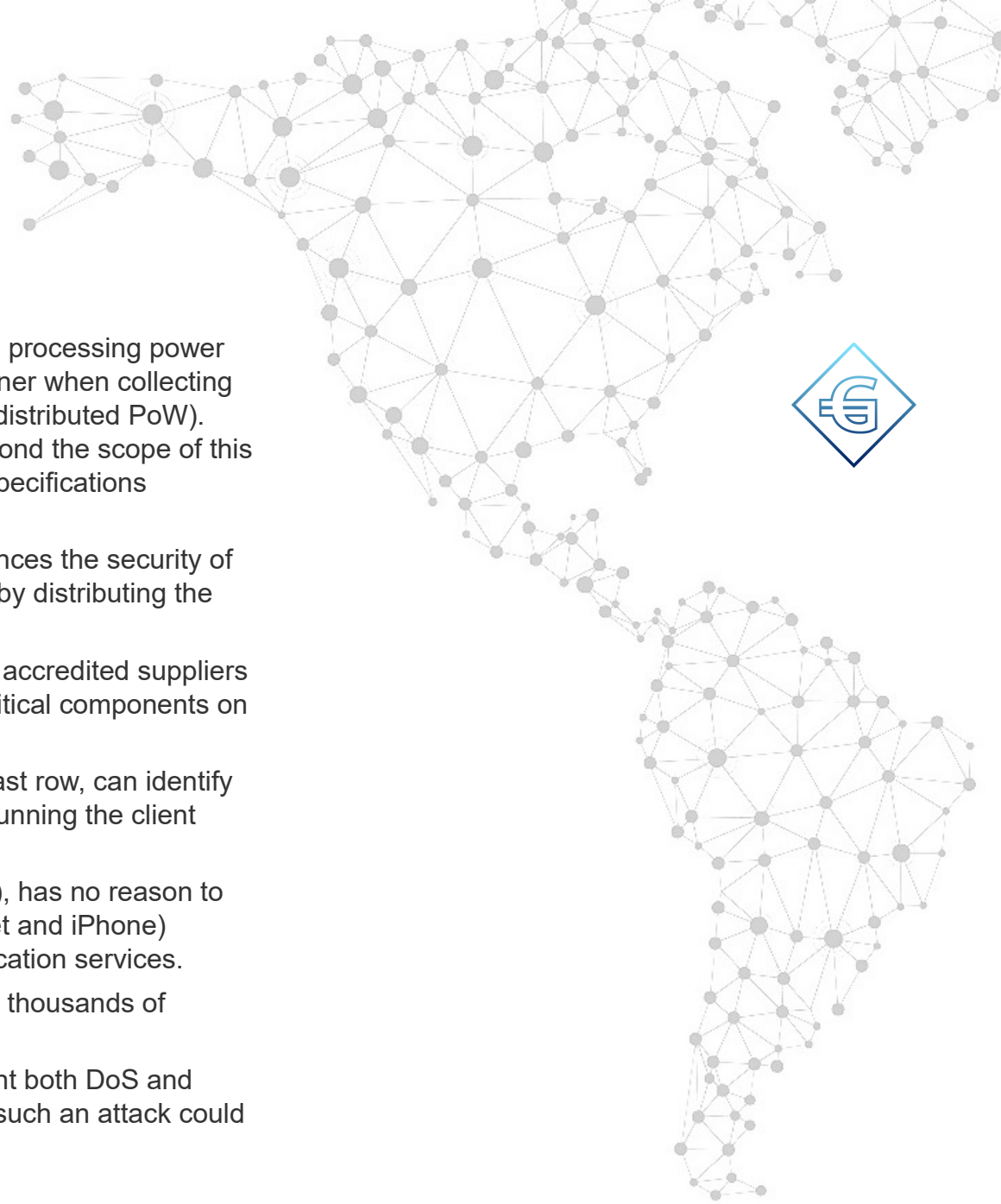
Current crypto-networks based on the Nakamoto design avoid this type of attack by requiring each node participating in the contest for block distribution (and corresponding reward) to prove its processing power (PoW). The idea is that simulated nodes, owned by the same person/entity but with a different address, also have to prove their processing power in order to compete. That is, the attacker needs twice the real processing power to have double the chance of winning the reward.

For a majority attack to have probability of success, an attacker would have to secure (by hacking, acquisition, co-operation or simulation) the majority of the processing power in the crypto-network.

While in current PoW networks the difficulty of such an attack is increased in proportion to technological advances (i.e., the speed of new ASIC processors), there are no mechanisms to predict or exclude such an attack.

2.2.1 Identity Verification

New-generation networks[2] introduce new hardware/software and procedures that can be used to better cope with malicious attacks involving duplication or simulation of identity. They also distribute processing power, control and the responsibility for security.



In new-generation crypto-networks, the proof of a node's processing power (PoW) instead of being demonstrated by one random miner when collecting a reward, is demonstrated by all nodes at all times (i.e., distributed PoW). The reasons and implications of such a decision are beyond the scope of this document and can be found in the Gorbyte Functional Specifications document.

Here we are interested in an additional aspect that enhances the security of the network: the prevention of DoS and majority attacks by distributing the responsibility for security.

This is accomplished by involving other entities, such as accredited suppliers and payment companies, in the process of registering critical components on the blockchain.

An address, used as described in Section 2.2, Table 1, last row, can identify a crypto-network node, which is associated to a device running the client software and owned by a user.

An honest user (or an honest employee of a corporation), has no reason to own more than a few personal devices (e.g., a PC, Tablet and iPhone) in order to use financial transaction and distributed application services.

A malicious user may instead attempt to own or simulate thousands of addresses, in order to mount an attack.

It is now possible to detect and prevent attempts to mount both DoS and majority attacks on the network much earlier than when such an attack could successfully occur.

This is achieved by using the concept of a node address to device association (NADA) between a crypto-network node and a virtual or hardware BRUD device.

First, the uniqueness of the device must be guaranteed. This is possible because each device has its unique identification and encryption keys.

This information is registered on the blockchain. The uniqueness of the association between the node address and the device (NADA) is verified at the time a node starts and wants to take part of the crypto-network as a full node.

At any time a node X wants to establish a session with a peer node Y, the receiving node Y can verify the device registration record on the blockchain and assure itself that X is not a clone and it can safely cooperate with X.

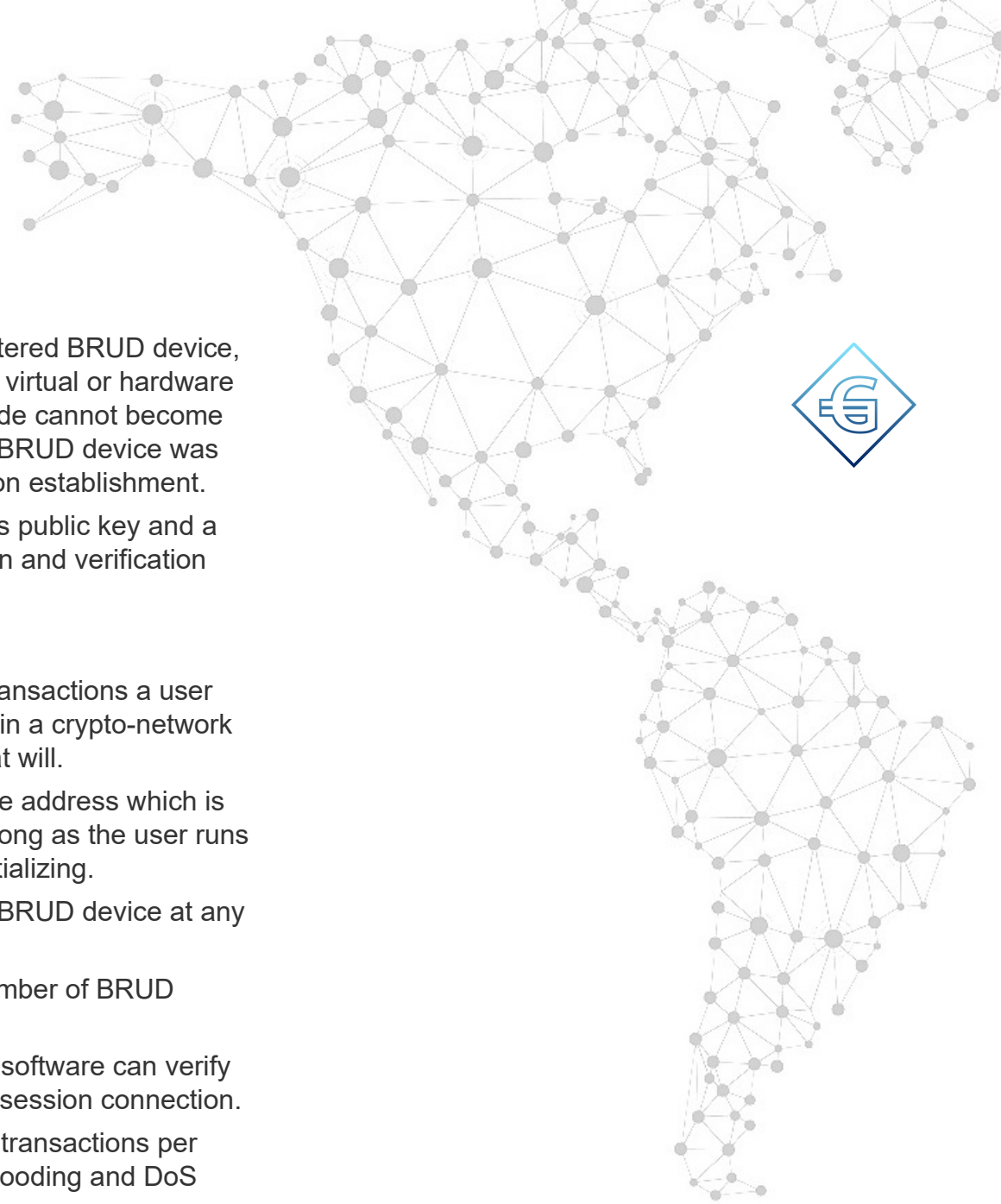
The protocols employed for these purposes are described in Section 3 of this document.

2.2.2 The Gorbyte Random Network

The Gorbyte crypto-network is implemented as a random network of G-nodes over the internet[3]. Each Gorbyte node can be identified with one address generated by a user.

This address is used as a node identifier when a Gorbyte user node initializes. During this initialization phase a G-node will automatically establish a number of session connections to peer nodes[4].

The verification of the NADA association is required every time one of these sessions is established.



The client code (GCC) must be running either on a registered BRUD device, or on another device (PC, iPhone, etc.) with a registered virtual or hardware BRUD device associated to it. In all cases, a Gorbyte node cannot become functional (a session cannot be established), unless the BRUD device was registered and is verified as original and unique at session establishment.

The verification of a BRUD device is done by checking its public key and a *datapack* of unique information. BRUD device registration and verification procedures are explained in more detail in section 3.

Preventing denial of service attacks

A way to prevent DoS attacks, is to limit the number of transactions a user can generate per block. Such a limitation is not possible in a crypto-network where a user can own real or simulated network nodes at will.

However, in Gorbyte, the user can be identified by a node address which is as persistent as the G-node itself. That is, it remains as long as the user runs a G-node and issues financial transactions without re-initializing.

A user can have only one node address associated to a BRUD device at any one time.

Furthermore, a user can purchase only a very limited number of BRUD devices (See sections 2.3 and 3.4).

Because of these verified restrictions, the Gorbyte client software can verify the identity and uniqueness of a peer node requesting a session connection.

The Gorbyte client software can also limit the number of transactions per user, per block to a small number, thus avoiding penny-flooding and DoS attacks from any one user.

Preventing majority attacks

A way to prevent majority attacks is to limit the number of nodes a user can run. The creation of multiple user identities (i.e., the ability of one device and/or person to act as a multitude) must be detected and prevented as soon as possible.

Multiple identities are used in brute force attacks, such as the hacking or acquisition of multiple nodes, and attacks that involve the automatic, programmed, simulation of multiple network nodes.

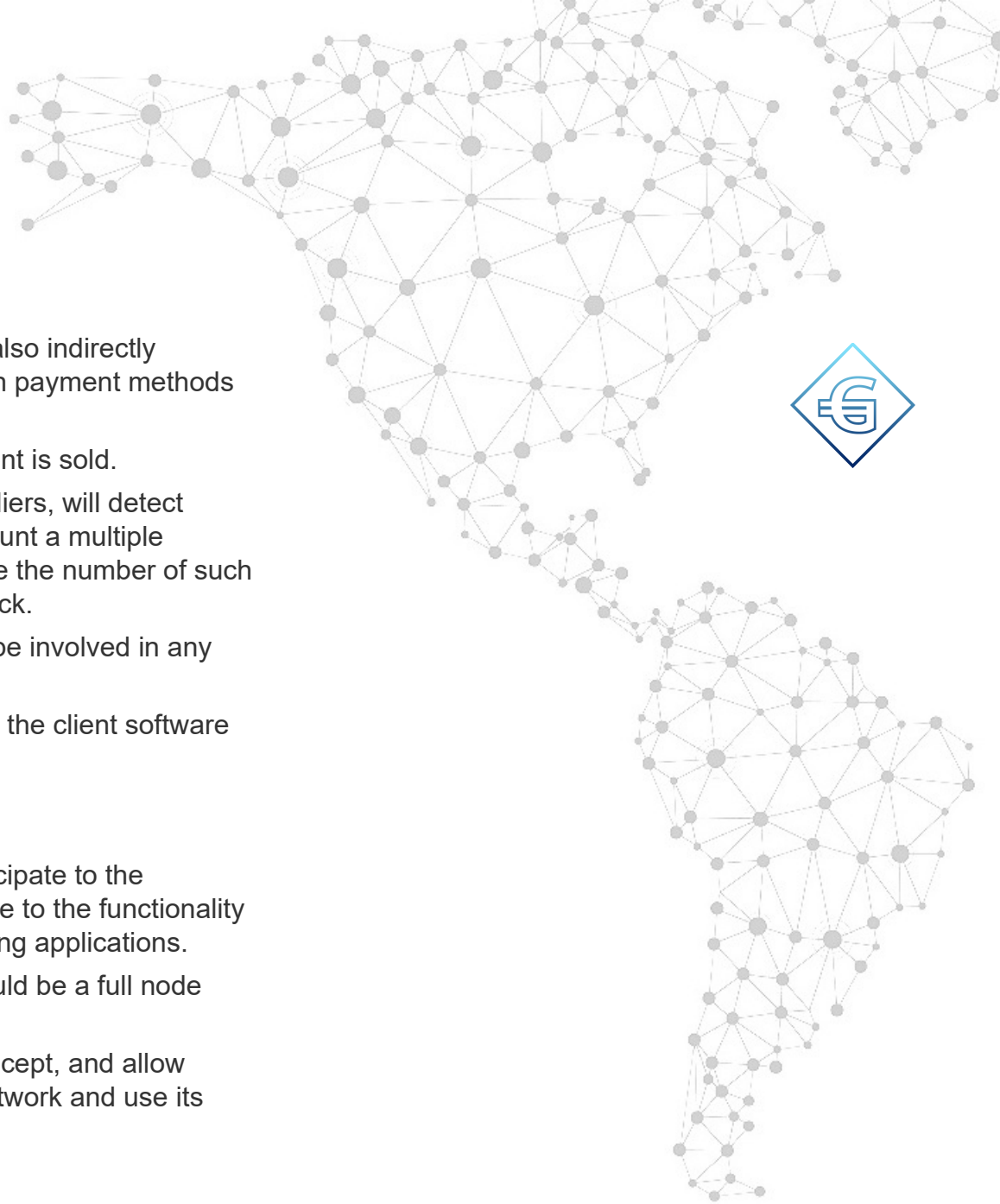
Such attacks, if successful, would be devastating to the network[5]. Current crypto-networks defend themselves from such attacks by increasing the difficulty for such attacks to succeed, but no warning signs are evident when the attack is in progress.

In Gorbyte, multiple identities are prevented by design through the use of blockchain-registered unique devices.

2.3 Distribution of Responsibility

As seen in the previous section, new crypto-networks rely on the distribution of responsibility for protecting against the proliferation of identities, to reduce the risk of attacks.

A step in this direction is accomplished by the concept of a BRUD device and by involving accredited BRUD device suppliers in the initial registration of the device on the blockchain. This is described in section 3.2.



As a consequence, accredited payment companies are also indirectly involved in the registration process, by providing limits on payment methods used for the acquisition of a BRUD device.

This is to guarantee that only one unit per form of payment is sold.

By setting this limit, payment companies as well as suppliers, will detect possible malicious parties trying to buy many units to mount a multiple identity attack. Such attempts will be detected well before the number of such units can reach the thousands needed for a majority attack.

However, neither suppliers nor payment companies will be involved in any other way after registration.

The verification of the device uniqueness will be done by the client software of peer nodes receiving session establishment requests.

2.4 Distributed Crypto-network Users

Distributed crypto-network users are full nodes that participate to the consensus process by majority agreement and participate to the functionality required for network security and for distributed processing applications.

Bitcoin started as a network where anyone with a PC could be a full node (a miner).

New crypto-networks (e.g. Gorbyte) re-establish that concept, and allow anyone with a PC to become a full node of the crypto-network and use its services.

2.5 BRUD Device Uniqueness

BRUD devices are an integral part of Gorbyte. In addition to adding to the security of the crypto-network, they provide a guarantee of uniqueness.

A virtual BRUD device uniquely identifies a crypto-network **node** through the node address to device association (NADA).

A real, tamper-proof, biometric BRUD devices can be used to guarantee the uniqueness of **users**. This is made possible by the association between a BRUD device (bonded to a user through biometrics) and a crypto-network node address.

In both cases a BRUD device can be registered on the Gorbyte public blockchain, so that it can be verified by peer nodes at the time a Gorbyte node comes up.

One of the advantages of the guarantee of uniqueness of a user is the ability to support virtual private blockchains (VPB).

2.6 Support for Virtual Private Blockchains

Gorbyte's open source code can be copied by anyone. Just as with any other crypto-networks the source code can be used to create a separate blockchain (a clone).

However, Gorbyte also includes native support for multiple private networks sharing the same public blockchain. This is possible because of the concept of uniqueness of a BRUD device.

The Virtual Private Blockchain (VPB) feature is much more useful than a cloned network. It allows, users to be part of multiple private clubs without isolating themselves from the rest of the world.

For example, users can be trusted customers of a bank (or more than one), and, at the same time, be able to interact with other distributed application available on the public blockchain and with the IoT world.

Additionally, private financial institutions, with their own VPB, may exchange transactions with each other on the public blockchain.

The concept is analogous to multiple VPNs sharing the same internet, but at a higher architectural level.

Private institutions may be interested in using the Gorbyte network code as the basis for their virtual private blockchain, because its source code is free, and because there are no running costs for basic transactions.

For example, a bank could use a Gorbyte VPB to create its own distributed operating environment, with its own distributed applications.
In this way, the bank could provide access restricted to its clients to unique, competitive services over and above basic financial services.

2.6.1 Subscribing to a VPB

A user may want to subscribe to a specific institution's virtual private blockchain. To do so he/she may interact with the institution, providing his/her unique node address and, through secret messages, whatever other information the institution requires from the user, according to the institution's KYC policy.

As a result of this subscription, the institution will store the address of this user's node on the public blockchain. I.e. This node address becomes part of a private network address group.

From then on, any peer node can verify that this user is unique (through the BRUD device's signed message) and has subscribed to an institution's private network address group.

3. Summary Description of a BRUD Device

A BRUD device is used to ensure the uniqueness of the identity of a crypto-network user to a specific degree, depending on the device model.

It does this through the host crypto-network client, by providing encrypted information at registration and session establishment times. This includes its unique model/serial/version identifier and encryption keys. The crypto-network client software verifies analogous information coming from other peer nodes requesting a session connection.

3.1 BRUD Protocol Introduction

The BRUD protocol operates at a lower layer than normal transactions (financial transactions or DOE transactions). The messages exchanged to verify a NADA association are at *session establishment* time and not during regular transaction collection and reconciliation. These messages are not recorded in the blockchain, nor need to be recorded anywhere else.

3.2 BRUD Device Registration

The first interaction occurs between the BRUD device buyer and its supplier.

For virtual BRUD devices, the supplier can be an accredited crypto-network foundation and the fee charged for the virtual device can be a nominal fee.

For hardware BRUD devices, the supplier is an accredited manufacturer.

When a device is ordered and paid for, the supplier hashes the account information used to pay for the device (**h**).

The device is shipped by the supplier with its **initial** unique encryption keys (**ID_{pk}** and **ID_{sk}**).

When the unit is shipped, the supplier issues a registration transaction for the device containing its public key (**S_{pk}**) and a signed *datapack* message including the **ID_{pk}**, the model number and serial number or version number (**msv**) and the hash (**h**).

The *datapack* will be interpreted by the crypto-client code (e.g.: Gorbyte's GCC) to verify that the issuer of the registration transaction is an accredited supplier. If so, the client code will verify that the **datetime** value is current and will store the registration information on the blockchain.

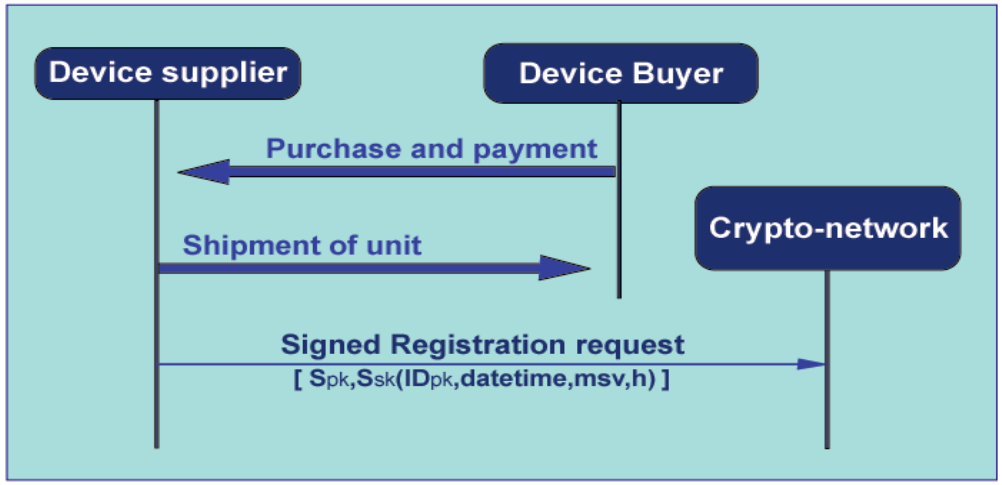


Diagram 1: BRUD device Registration

Diagram 1 above, shows the sequence of registration interactions between the BRUD device supplier, the buyer, and the crypto-network (i.e. the client code running on each crypto-network node).

The BRUD device registration procedure is a particular instance of a more general registration procedure for any hardware or software component in the crypto-network, which is detailed in a separate document.

3.3 Registered Key Change

After the device registration process is completed, the supplier will not need to be involved in the process of verification, or in any other way.

A *Registered Key Change*, issued by the BRUD after it becomes operational, adds an extra level of security. Its objective is to cut off any dependency from the supplier and prevent possible malicious attacks from the accredited supplier if it turned malicious, or from an attacker that could have been able to acquire the supplier's data pertinent to encryption keys.

Any other information known by the supplier is now public and immutable on the blockchain (i.e. **msv** and **h**).

A BRUD device can change its encryption key pair, but cannot change its other original registration information.

Registered Key Change transactions are also stored on the blockchain. At the time of verification by peer G-nodes, the public key used is the one stored by the last *Registered Key Change* transaction ($\mathbf{D_{PK}}$).

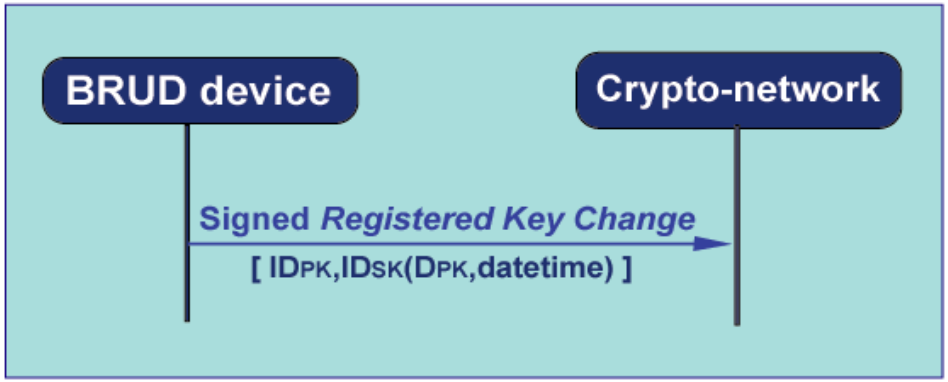


Diagram 2: Registered Key Change transaction

ID_{PK} is the original, or previously registered, device public key hash.

ID_{SK} is the original, or previously registered, secret key used to encrypt the message.

The BRUD device *Registered Key Change* procedure is also a particular instance of a more general *Registered Key Change* procedure for any hardware or software component in the crypto-network, which is detailed in a separate document.

3.4 BRUD Device Verification at Connection Time

A new crypto-network client node, running on a device with a virtual BRUD or running on a BRUD device, will associate its node address to the device (NADA association) when requesting new connections, to become functional.

This is done during the session establishment phase. A session connection request includes, in addition to lower level addressing, session specific parameters. It also includes the sender node address, the current datetime value, and the other signed information about the device identity.

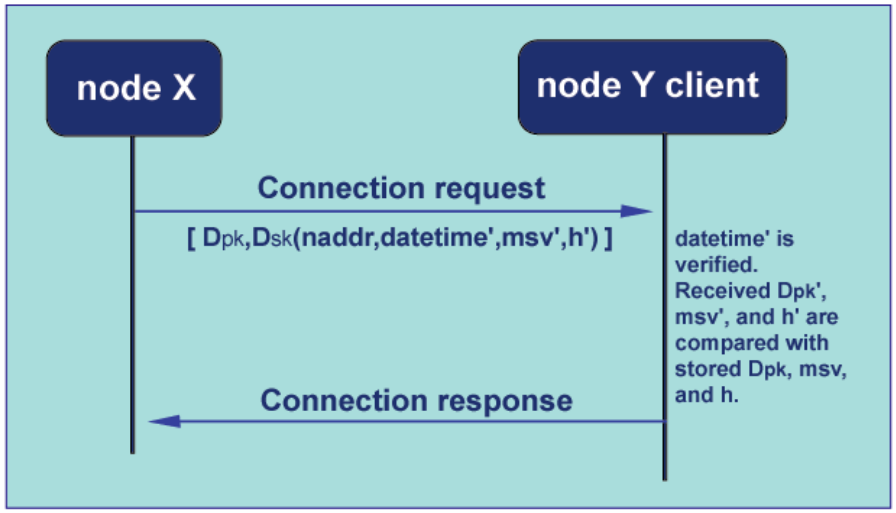


Diagram 3: BRUD device verification at connection time

Diagram 2, shows how a session connection request includes a *datapack* signed by the device with its secret key ($\mathbf{D_{sk}}$). This signature guarantees that the encrypted *datapack* comes from the device with public key $\mathbf{D}_{pk}$.

The receiving node uses the $\mathbf{D_{pk}}$ to decrypt the *datapack*. Then it verifies that the message time is within the required limits. The current datetime guarantees that the *datapack* is always unique and not re-used. Thus the receiving node knows that only the device with that public key ($\mathbf{D_{pk}}$) could have encrypted the *datapack* and it did so only a few seconds ago.

The receiving node then looks for the last transaction record in the blockchain with identical data.

If this is found, this is an assurance that then the received *datapack* is from one authentic and unique device that has been registered. Thus the receiving node can trust that the node address in the *datapack* (**naddr**) is the one associated to that device, and can accept the connection.

This guarantees that, for the duration of that association, that node address is used by the unique individual who owns that specific, verified device.

The requesting node can then ask for new node associations using the same BRUD device, as it must connect to a number of logical neighbors. However, the number of such connection is limited, for all nodes, by a crypto-network global parameter (Gorbyte's **Max_s**) that can only be changed through a client code version change.

In Gorbyte, global parameters cannot be changed unilaterally by one node without disqualifying the node: such change would not pass peer verification during protocol interactions.

Note that more than one BRUD device can be associated to the same node. This can be done by establishing new connections to new logical neighbors using different BRUD devices. However, as we have seen, a user is severely restricted in the number of BRUD devices he/she can buy and connections it can establish. Furthermore, this configuration may or may not be handled seamlessly by higher level general distributed applications.

In addition to functionality enhancing security and verifying the uniqueness of a node, future BRUD devices will include other functions.

The evolution of the functionality of BRUD devices is described in the next section.

3.5 The BRUD Device Evolution

The BRUD device is initially implemented as a virtual device: A software component running on a host device connected to the internet.

This virtual device provides uniqueness functionality, as we have shown in the previous sections.

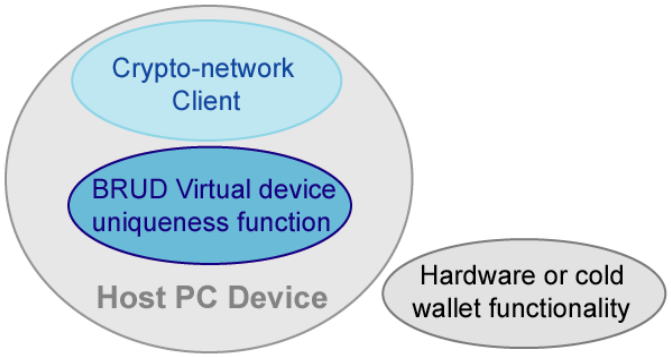


Diagram 4: First Stage:The BRUD Virtual Device

Although this is a good initial solution, a more functional and secure solution can be implemented on a hardware device. For this reason, the BRUD architecture defines an evolution path, from virtual, to firmware, and finally to a wearable, tamper-proof, biometric device.

In the virtual device implementation, the secrecy of the virtual device private key must be guaranteed by the owner, by protecting access to his/her host PC system with current access control methods.

More advanced BRUD device models will rely instead on the tamper-proof qualities of the device itself.

In addition, more advanced BRUD device models will include new hardware and software functionality, such as biometrics, and will include functionality that was initially running in the host PC.

The evolution of such solutions may go through one or more of the following stages.

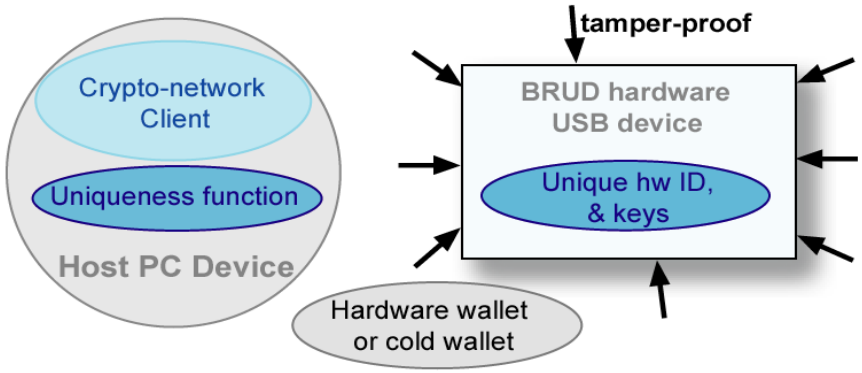


Diagram 5: Second Stage:The BRUD device as a USB device

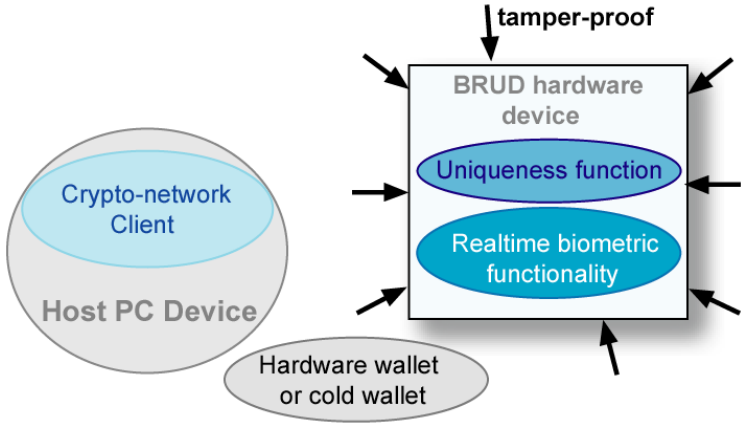


Diagram 6: Third Stage:The BRUD device as an intelligent device

In the above diagram, an intelligent and self-contained function has been added to the BRUD device: a biometric functionality able to recognize the uniqueness of the user and prove that the user is alive.

Future models will include new, original functionality, such as accessing general distributed applications (GApps) written for new crypto-networks.

New BRUD device models may also implement hardware wallet functionality.

These upgrades will eventually allow the user to conveniently and securely use all the services he/she needs from its BRUD Device.

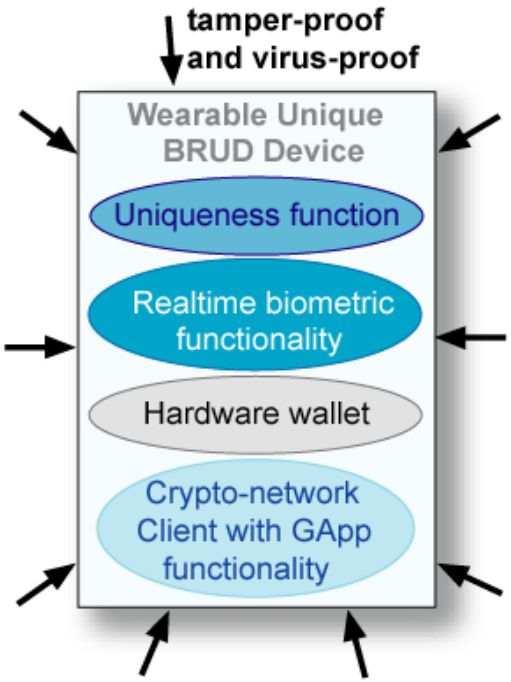


Diagram 7: Final Stage:The BRUD device as a DOE device

The Final stage model of the BRUD evolution, the tamper-proof, wearable, biometric BRUD device, will provide the user with access to the crypto-network Distributed Operating Environment (DOE).

The DOE is conceptually en extension of a current operating system, where the problems of security, communication, identification, data integrity, and distributed processing, have been solved compatibly for all applications.

Once the crypto-network client software is included, the software in the device itself will be guaranteed to be theoretically virus-proof: Each software component of the device, from the initial bootstrap and kernel, to the internet stack, to the various GApps will be hashed. Their unique hash, version number and code will be distributed and signed by the entity that developed that component, so that malicious alterations can be immediately detected.

The requirement for a secure software stack for BRUD devices will create new opportunities for crypto-networks and software companies to compete in developing a verifiably secure software environment for their users.

Thus the BRUD device will allow users to carry only one secure device for all of their requirements of blockchain-based functionality, such as identification, authorization, communication and payment needs.

For example,BRUD devices will be used for secret and secure messaging, for voting at elections; when paying for a meal, or buying airline tickets; for financial transactions, or for E-commerce; for buying/selling assets; for accessing airport security, work environments, or entertainment venues; for accessing medical records and emergency information; for proof of licenses, proof of ownership, etc.

APPENDICES

A Digital Access and Identity Classification

The distribution of identity authorization, and citizen's ownership of identity, can be gradually accomplished, in accordance with the classification presented below, or directly by implementing the BRUD PoPI model, described in Section A.2.3.

The above will depend on the improvements in biometric capture and techniques in the next few years.

Before talking about identity, we must consider digital access control. Most of the times, this has to do with protection from attacks from human or programmed threats: BOTs, Access to private accounts, Denial of Service attacks, and Multiple Identity attacks.

It is possible to classify identification according to a number of traits that together form the digital identity of a real person. Various such classifications are available in the literature.

For example, the use of an identity can be described by the number of times a unique identification is required (E.g., anonyms, anonymous identifiers, pseudonyms).

By unbundling the various traits of someone's identity we can provide some level of privacy by revealing only a subset of the traits associated to a person. This allows for controlling to what degree an identity is revealed and/or what degree of privacy is maintained.

A.1 A Progressive Scale

Here we distinguish access and identity and introduce a progressive scale of access control, authentication and authorization.

Access/identification	Technique	Comments
No specific approach with respect to access or identification.	General security of ISP, PC, Email, anti-virus, anti-malware, firewalls.	If any personal information is disclosed, this is not secure.
Pseudonym/account name & password created for a web site.	May include riddle, but no other security technique.	Relies on the web site security.
Proof of Human Origin (**PoHO**). Only biometrics information is disclosed.	Biometrics (e.g. iris, fingerprint, etc.) with no corroborating verification.	Reproducible, copyable.
Proof of Humanity (**PoH**) e.g., the person's DNA is disclosed.	Biometrics (e.g., DNA) with no corroborating verification.	Some biometrics may be non-reproducible, some may not be copyable.
Safe Pseudonym /account name & password created for a web site.	Includes SSL connection, but no ID is requested.	Relies on the web site security.
Access to a personal bank account through a card.	Debit card PIN, CVV, chip	Relies on the Bank security.

Access/identification	Technique	Comments
Safe callback pseudonym/ account name&password created for a web site.	Includes SSL connection and call back to a phone number, but no ID requested.	Relies on the web site security. The phone number is not quite an anonymous identifier.
Proof of device uniqueness (**PoDU**)	Combination of a unique hardware, software and encryption (Basic BRUD). No biometric or personal information is collected.	Initially used by Gorbyte to reduce Denial of Service, Multiple Identity, and Simulation attacks.
Proof of Unique Human Origin (**PoHU**).	A device capable of capturing a biometric pattern, including verification of human origin. Only this pattern is disclosed. No personal ID information is disclosed.	This is used once as an anonymous identifier. A hacker cannot prove PoHU without using a ***real person biometric***.
Proof of Unique Live Human Origin (**PoHL**).	A device capable of capturing a biometric pattern, including verification of a live human. Only this pattern is disclosed. No personal ID information is disclosed.	This is used once as an anonymous identifier. A hacker cannot prove PoHL without being or using, a ***real live person***.

Authentication, authorization and digital signatures	Technique	Comments
After account is created, credentials are asked and entered on a web site form.	Form to be filled using SSL access. May include a call back to a phone number.	Relies on the web site security.
Protection by Digital identifiers (account)	Creating an account, using Password, Email, PIN, etc. with multiple trusted Entities (e.g., Banks, Credit cards)	All elements in the account are protected by a trusted entity, but hackable
Submitted Proof of unique identity (**PoUI**)	Some standardized system keeping ID info in a secure place.	Unique identity has been proven by a trusted third party.
Unique Identity authentication (**PoIA**)	Some standardized system keeping ID info safe, but providing public access.	Anyone can see the proof of unique Identity without trusting a third party.
Citizens' owned and controlled Private Identity (**PoPI**)	The device can uniquely recognize its owner at any time. No other person can claim its ownership. See Section	The owner of the Identity can prove it to anyone without disclosing personal information.

A.2 The BRUD Device in the New Crypto-network Environment

Out of the above classification, some of the identified levels can be implemented by different device models (PoDU, PoHU, PoHL, PoPI).

Each Gorbyte node shall support a BRUD device (initially a virtual device). Gorbyte may require updates of the device to support its future distributed frameworks, as technology improves.

Virtual BRUD devices will be provided as Open Source code.

The functionality of BRUD devices may also be extended, as technology solutions in biometrics evolve.

The advanced BRUD device models maintain user access and identification information, and encryption keys in a controlled, standardized and tamper-proof hardware environment.

For example, Trusted Cryptographic Module (TCM)[6] techniques are currently being investigated for this purpose, and similar techniques may be available in practice in the next few years.

Eventually BRUD devices will include various functionalities, as seen in section 3.5.

The BRUD_PoPI, described in A.3.3, will allow for a simpler development of a range of blockchain applications that require secure authorization.

A.3 BRUD Device Evolution from PoDU to PoPI

The reasons for using a BRUD device, as mentioned in Section 1.3, are various, and include the distribution of responsibility for protecting against the proliferation of identities, by involving the device suppliers in registering of their products on the blockchain.

The number of BRUD devices a user can buy is limited. The supplier will only sell one device per form of payment.

The restriction based on the form of payment is introduced so that, in addition to the supplier, other corporate entities can be alerted in those cases where a malicious attacker may attempt to purchase a high number of devices from different accredited suppliers. Reputable payment companies are major credit card companies, major banks, PayPal, Western Union, etc.

It is important to notice that neither the suppliers nor the payment companies have anything to do with the verification of the device, or the verification of transactions. Their only role is to register a unique device on the blockchain.

The association to a node address (i.e., the NADA association) will be verified by peer nodes. The verification process is done by any node client software responding to a session connection request.

The crypto-network would, in principle, still work without BRUD devices, with its security guaranteed by its distributed PoW. However, as mentioned earlier, DoS and majority attacks would be more difficult to detect and curb at their very start.

BRUD device models can be classified according to their ability to deliver higher levels of access control and personal live human authentication and authorization (as seen in Appendix A.1). A different classification and evolution according to functionality was provided in section 3.5.

A.3.1 The BRUD_PoDU Model

The first version of the BRUD device to be used by Gorbyte (BRUD_PoDU) does not maintain user information. Thus, its privacy is intrinsically the highest possible.

This device model can be a virtual device, which is downloadable from a crypto-network foundation web site (e.g. GorbyteFoundation.org).
The foundation is responsible for the registration of each virtual device version on the blockchain.

This first device model is only a means for assuring the uniqueness of the NADA.

The unit's identification information (e.g: a model and serial number selected and assigned to the unit by the supplier) assures the unit's uniqueness.

The encryption keys assigned to each BRUD device are used to sign and verify the registration of the unit on the blockchain and to verify the origin of messages from the device. Even if these keys were somehow hacked, the attacker would only be able to use this unit as one device. The attacker would not be able to duplicate its identification for multiple fictitious units, since its validity (existence) and its uniqueness are assured by the

registration information on the blockchain, originally created and signed by an accredited supplier.

The supplier of this first virtual device model would need to sell a unit for a nominal fee and register it on the blockchain, according to the BRUD protocol and procedures described in Section 3.1.

A.3.2 Future BRUD Device Models

Future models of the device (i.e., models above the first BRUD_PoDU) will store and maintain user information.

Future models of device with advanced biometric capabilities will use the latest tamper-proof hardware technology to guarantee user privacy.

In case the BRUD device is lost or stolen, it cannot be used by a new person as the old biometrics will not match any other user.

In case the unit is lost, stolen or becomes nonfunctional, the original can be replaced by a new unit, which will be re-registered on the blockchain (bound to the person). The new unit Keys will need to be changed at the time it is received.

A user is still limited through the registration and verification procedures:

- to using (binding) only one unit at a time; and
- in the number of units that a user can buy.

In case the unit needs to be accessed for forensics purposes, the owner is encouraged to cooperate.

Access to the unit will only be allowed through the courts (e.g., a warrant from a judge).

The unit is used for proof of personal identity only. It cannot guarantee the legal behavior of the owner nor guarantee the legal use of the identity. For example, a member of the owner's family could be held for ransom, to force the owner to perform a financial transaction.

A.3.3 The BRUD_PoPI Model

This model uses existing state of the art technology to implement a citizen's-owned and controlled Private Identity, which conforms to the current UK Payment Service Directive[7].

Here we describe only the basic functionality of the device related to the uniqueness level of identity required by a crypto-network. The device can also be used for holding other personal information that needs to be maintained both securely and readily available to the user.

The level of reliability of a unique identity is determined by the strength of the biometric measures used by the device, presumably increasing as technology improves.

The following are the four factors[8] the owner needs to match, in order to access the device, and prove his/her unique live identity:

Information the user **knows**	A PIN or Password (The encryption keys are only used by the device to sign messages).
A device the user **owns**	The device itself (Identified by model/serial number).
A feature showing who the user **is**.	A biometric measure (face, iris, fingerprint, voice, finger movements, etc.)
An action the user must **do**.	A random action, performed in real time and related to the above feature, as a confirmation of *being alive*.

The random action would be related to the type of biometrics used. For example, if voice recognition was used, the random action could be: *requiring the user to repeat a random phrase prompted by the device*. However, the level of reliability can be increased by using a combination of multiple biometric techniques.

The random action is used to prevent spoofing. That is, the duplication of the owner's biometric information to cheat the process.

When the user can match the above factors, he/she can prove to be the owner and a live person to a predetermined level of reliability.

The BRUD device itself can prove its uniqueness by signing a message with its identification information. Thus, the combination device-Owner prevents spoofing to a certain level[9].

The above process still leaves open the possibility that the owner could be forced, by a criminal, to provide the above factors under threat or as ransom.

However, from the point of view of the new crypto-network requirements, this possibility still cannot help a criminal to mount a majority attack or a DoS attack, since he would have to repeat his criminal exploit many thousands of times, with different people, within a limited time-frame.

In conclusion, a BRUD_PoPI device built with today's technology could be used for the purposes of new crypto-networks, such as Gorbyte.

B Manufacturers' Opportunity

As mentioned, manufacturers will sell BRUD devices for a price. This will introduce opportunity and competition. More importantly this will also introduce a stake for manufacturers in the crypto-network business.

Manufacturers will also profit from continuous developments in the functionality of BRUD devices (As delineated in sections 3.5 and A.3) and consequent future opportunities in the biometric and identification fields.

To participate in the BRUD device manufacturing business and become accredited by a crypto-currency foundation, manufacturers are required to exhibit a high level of responsibility and openness.

BRUD devices will be provided by several competing manufacturers. Manufacturers shall be accredited by the Gorbyte Foundation and possibly by other crypto-networks. The list of accredited manufacturers will be listed in these public web sites.

The Gorbyte Foundation and possibly other organizations will monitor the sale of devices by different manufacturers and may remove a manufacturer from its accredited list, if the manufacturer does not maintain its standards.

Furthermore, manufacturers are publicly known and are accredited only from those countries that commit to prosecute illegal activities, so that their owners and directors could face legal consequences.

C BRUD Device General Requirements

A BRUD device supplier needs to comply with the crypto-network specifications, the BRUD specifications described in Section 3.1, and the following requirements:

Procedural Requirement	Virtual device	Future devices
BRUD devices will be sold by manufacturers and software suppliers for a fee (possibly a nominal fee).	✔	✔
Buyers of BRUD devicess will have their form of payment hashed by the supplier in order to have their identity stored in the blockchain.	✔	✔
The device and its procedures shall be certified by the Gorbyte Foundation before being released.	✔	✔
The supplier is required to supply an initial encryption key pair, to be used in conjunction with Gorbyte for the registration of its identity information on the blockchain.	✔	✔
The device will evolve, from model to model, according to state of the art technology in the biometric, security and identification fields.		✔
The supplier name and its current public key will be included in a public list of accredited suppliers on public crypto-network sites.	✔	✔

Table 2: Procedural requirements of BRUD device suppliers

Software/Hardware Requirement	Virtual device	Future devices
The BRUD device shall not be operational until the owner has registered it with the supplier at the time of purchase.	✔	✔
Any hardware/firmware of the unit shall be tamper-proof. If the device or the private encryption key is tampered with, the unit shall become inoperable.		✔
The device shall evolve towards its final model, which will contain all the functionality a user needs to take advantage of the services of new crypto-networks and their GApps, running on a distributed operating environment (DOE)		✔
The BRUD device shall be identified uniquely (e.g., model number, serial number) and unique encryption keys.	✔	✔
A client node shall accept only a maximum of one transaction request per minute from the same node addresses.	✔	✔
A client node, receiving a session connection request from a peer node, can verify that the unit, with that public key and identification data is registered on the blockchain, thus it is not a simulated unit.	✔	✔

Table 3: Software/Hardware requirements of BRUD devices

D Verifications Made Possible by BRUD Devices

The following are general verification steps common to all BRUD models. Specific verifications will be described in the Specifications document for each BRUD model.

The Gorbyte node address / device association (NADA) can be shown to be unique because of the following:

1. The device used is unique:

The *datapack* includes supplier's data: The signed *datapack* is decrypted using the supplier's public key (S_{pk}). The Model number and Serial number, or version number, and hash of the device are compared with the same data in the last blockchain registration transaction.

The supplier's public key can be verified against the supplier's site and any other real-time site that maintains a list of accredited manufacturers with their public keys.

2. The PoDU datapack was not duplicated:

The *datapack* includes an absolute datetime stamp[10]. Since the message is signed and can be verified, the datetime value can also be verified by any receiver of a session connection request.

3. *Session establishment verification:*

At the time of session establishment, each nei-peer will verify the *datapack*, before it accepts the establishment of a connection. The nei-peer decrypts the *datapack* (received from a peer node requesting the establishment of a session) with the BRUD device's current public key. Once the message is in the clear, the nei-peer will verify that the device identity information in the *datapack* matches the information on the blockchain. If so, the receiver can trust the node address provided by the sender in the *datapack* (naddr). This address cannot have been corrupted "on the way" from the node requesting the connection to the node receiving the connection request.

To prevent copies of the *datapack* from previous transactions, the *datapack* contains a time stamp (datetime). This guarantees that *datapacks* have a unique encrypted pattern.

The client code of the receiver node also verifies that the datetime is recent, by comparing it with the current absolute time (e.g. must be less than 5 seconds old). If one of the above verification fails, the receiving node will not accept the session connection request.

4. *Transaction per block limit:*

The user (Gorbyte client) cannot initiate more than a maximum number of transactions per block from the same address (e.g., two per block, more or less equivalent to limiting the number of transactions per person to one per minute). Since the address cannot be duplicated without an associated BRUD device, this discourages Denial of Service attacks.

5. *Limited opportunity to buy multiple BRUD devices:*

A hash of the *form of payment* method of each BRUD device when it was bought is also included in the information registered in the blockchain. This is a unique hash, thus it does nor reveal the original payment information.

By verifying this hash, the receiver node contributes to restricting potential attackers form buying multiple BRUD devices. This restricts the ability of a single buyer to mount a multiple node attack.

E Hacking the System

The potential attacks described in this section describe the security features and the resilience to attacks, added by BRUD devices, over and above the basic security of distributed consensus crypto-networks guaranteed by other means, including the Distributed PoW.

The details of how the Gorbyte crypto-network uses its Distributed PoW to reduce the possibility of malicious attacks is included in the Gorbyte Functional Specifications document.

Crypto-network software and hardware modules are also protected against attack by component registration and verification procedures that apply to every component in the network. The device registration and verification procedures described in this document are a subset of those procedures.

In addition, since distributed consensus crypto-networks do not require miners and do not distribute rewards, the only remaining incentives for an attacker are provided by the possibility of hacking, double spending, or simulating a transaction or group of transactions.

These actions are protected by cryptographic security measures common to all crypto-networks. However, Gorbyte took another step further by introducing Vaults (See: Gorbyte Functional Specifications).

The summation of all these measures (BRUD devices, Distributed PoW, component registration, no miner rewards, and Vaults) reduces considerably the incentive for, and the possibility of, malicious attacks in Gorbyte.

Each malicious attack scenario is further described below.

E.1 Obtaining the Manufacturer's Private Key

If an attacker can get hold of a manufacturer's key, despite the security features described in Section 3 (SSL, manufacturer's signature) he/she could simulate new units with correct Model and Serial numbers and with the specific manufacturer's data encryption keys.

However, since new units are required to change their encryption keys as soon as they become operational, the attacker can derive no value from the stolen initial keys provided to new units by the supplier.

E.2 Buying Most of the BRUD Devices

An attacker might plan to buy multiple BRUD devices to eventually own the majority of such devices in a crypto-network. This would be very difficult because of the limitation imposed by each BRUD device supplier during the transaction required to buy a device. That is, each customer can only buy one unit using the same form of payment (e.g. credit card, debit card, Paypal, bank account, Western Union, etc.). The manufacturer can also limit the types of payments it accepts.

To be successful, the attacker would have to spend a lot of time and effort to acquire thousands of valid credit cards, or convince a large number of people to participate in his illegal activity.

Finally, if this could be accomplished over a long period of time, the attacker risks owning thousands of obsolete devices since, from time to time, the manufacturers will introduce new BRUD models to the market.

E.3 Stealing the Private Key by Tampering

An attacker could tamper with a BRUD device to steal its private key with the intention of simulating multiple devices, all using the same private key.

This would not work, because his simulated devices have not been registered on the blockchain and they will not pass the verification checks from peer nodes.

If the attacker wanted to register a BRUD device, he would need the complicity of an accredited supplier to create and sign registration request with the supplier's secret key. He would have to do so for each stolen device, to create a fake identity on the blockchain for each one of them.

The attacker could possibly be able to create more transactions than generally allowed, but could not mount a DoS attack or a majority attack.

In addition, the stealing and/or tampering of multiple devices, if achievable at all, would be noticed much before the attacker could collect a large number of them.

E.4 Stealing or Hacking most BRUD Devices

If an attacker manages to steal or hack thousands of individual BRUD devices and use their data and keys, such units will have to be stolen from different people.

The attacker would have to mount his massive attack on the network before some of the real owners report the theft or report being hacked.

Alternatively, an attacker would have to pay thousands of people for them to participate in the malicious operation. These people would have to remain silent after being paid. The attacker would have to do this within a limited time, before the scam is discovered.

If the above could be achieved, and the attacker acquired control of most of the BRUD devices in the network, he still would have to contend with acquiring as many IP addresses for those specific nodes.

Finally, the attacker would face a similar problem as when mounting a majority attack on other crypto-networks. He/she would have to simulate, using real processing power (distributed Proof of Work), as many network nodes as the majority in the network. This activity would also be suspicious and would have to be done within a short time, without programming errors and with limited testing, before the other malicious activities are discovered.

E.5 Denial of Service Attacks

The node address is checked during transaction verification, to limit the number of transactions per address and per block to a set maximum (e.g., two). This limit is set to reduce the possible effects of Denial of Service attacks.

This limit is effective because an attacker will need to register one BRUD device per node address. The attacker could own or simulate thousands of addresses, in order to mount an attack, but even with such a setup, each node/device is limited to about a transaction per minute.

Thus such attempts to mount a DoS attacks would be detected much earlier than when they can be successful.

E.6 Client Software Simulated

An attacker could simulate the whole client node software.

Because of the Gorbyte component registration and verification procedures, such a client would not be able to establish sessions with nei-peers.

If a crypto-network did not have component registration and verification procedures, the attacker would be able to influence the result of a block only by acquiring/simulating more than half of all nodes of the network. In this type of attack the attacker would have to associate a BRUD device to each simulated node, encountering the hurdles mentioned earlier in this section.

E.7 Client Software Hacked

An attacker could attack user nodes' client software through a virus or other malware.

Because of the Gorbyte component registration and verification procedures, the modified software components would not be able to interact with other components in the same system.

If a crypto-network did not have component registration and verification procedures, the attacker would be able to influence the result of a block only by hacking more than half of all nodes of the network.

E.8 Multiple Identity Attacks

An attacker could use a combination of the above techniques, in ways not yet conceived, to present itself to the crypto-network with multiple identities.

Continuous improvements of the BRUD devices will specifically address multiple identity attacks. As in the above cases, the rewards to an attacker would be minor with respect to the costs. The highest incentives to the attacker (as we have seen with other crypto-networks) are in immediate cash-in by attacking individual user contracts, wallets and the weakest points in the network: usually user-written code.

As in the above cases, the attacker would have to associate a BRUD device to each simulated node, encountering the hurdles mentioned earlier in this section.

The details of how the Gorbyte crypto-network Distributed PoW, component registration and verification, and Vault mechanisms further reduce the possibility of these types of attacks is included in other Gorbyte documents.

Notes

[1] These parties are not involved in controlling, verifying or communicating with the BRUD device once these are in operation.

[2] New generation crypto-networks, are defined here as those public, unpermissioned crypto-networks that distribute the responsibility for their consensus and security mechanisms, such as Gorbyte.

[3] See: The Introduction to Gorbyte at: http://gorbyte.com/documents/Gorbyte%20Introduction.pdf

[4] The details of this phase are explained in the Gorbyte Specifications document.

[5] For example, if a crypto-network was taken over by a majority attack, much of its currency could be stolen and the value of its currency in the hands of honest users could drop considerably. After such an attack, in the best case the crypto-network could be segregated and survive as a hard fork; in the worst case it could cease to exist.

[6] See: http://ieeexplore.ieee.org/document/6598022/

[7] See PDF at: http://www.paymentsuk.org.uk/sites/default/files/PSD2%20report%20June%202016.pdf

[8] Identified by Steve Cook, Director of Biometric Authorization at Daon, London, UK.

[9] For our crypto-network purposes this level of reliability is much stronger than what is needed in order to prevent multiple-identity and DoS attacks. Gorbyte uses BRUD devices mainly to prevent the thousands of node address to device associations (NADA) that would be required, at the same time, to mount such an attack.

[10] See the *datapack* format in each BRUD model Functional Specifications.