Hydro Raindrop

Public Authentication On The Blockchain

January 2018

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

TABLE OF CONTENTS

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Abstract

HYDRO: Etymology - From Ancient Greek ὑδρο- (*hudro-*), from ὕδωρ (*húdōr*, "water")

Hydro enables new and existing private systems to seamlessly integrate and leverage the immutable & transparent dynamics of a public blockchain to enhance application and document security, identity management, transactions, and artificial intelligence.

In this paper, a case will be made for private systems, such as APIs, to use the Hydro public blockchain to enhance security through public authentication.

The proposed technology is called "Raindrop" - a transaction performed through a smart contract that validates private system access publicly, and can complement existing private authentication methods. The technology is intended to provide additional security for sensitive financial data that is increasingly at risk from hacking and breaches.

Initial implementation of the Hydro Raindrop is performed on the Hydrogen API Platform. This modular set of APIs is available to enterprises and developers globally to prototype, build, test, and deploy sophisticated financial technology platforms and products.

The Hydro Raindrop will be made available to the world developer community as open source software, to allow developers to integrate the Hydro Raindrop with any REST API.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Blockchain & Ethereum

Hydro is implemented on the Ethereum network. Before providing more detail on the project, it is important to understand some fundamental ideas about blockchain and Ethereum.

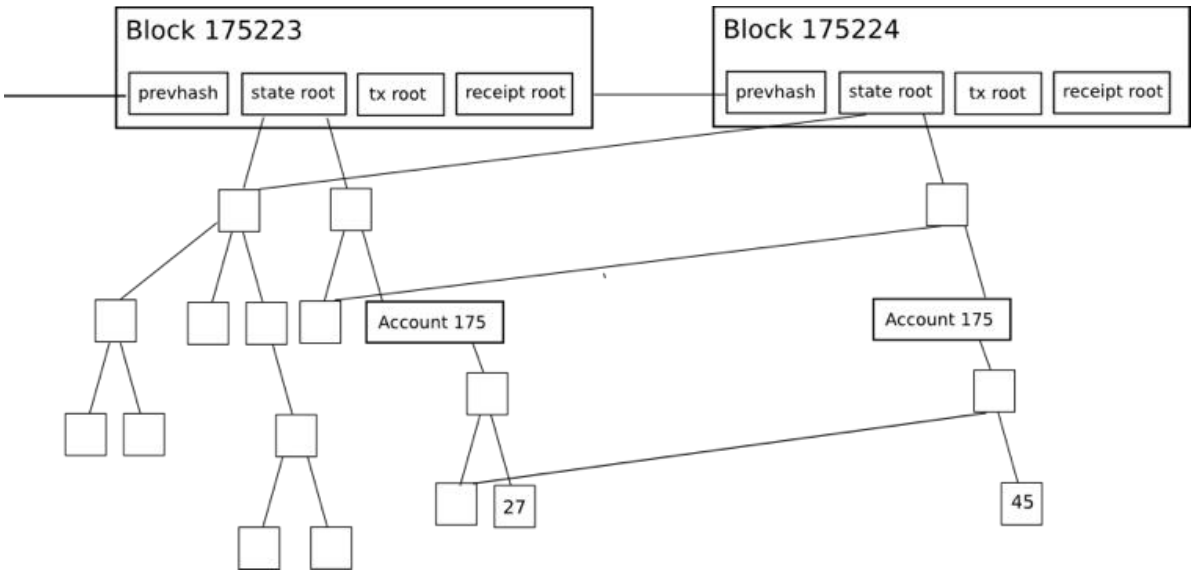
Building on Ethereum

Much as apps like Snapchat were built with Swift and other tools offered on top of the Apple iOS platform, so too can blockchain applications be built on top of Ethereum. Snap Inc. didn't need to build iOS, it used it as infrastructure to launch a game-changing social media application.

Project Hydro is similar. It relies on the thousands of developers globally that are working to make underlying blockchain technology faster, stronger, and more efficient. Hydro leverages this constantly improving infrastructure by developing product-focused interactions around blockchain technology that can offer tangible benefits to financial services applications.

Merkle Trees

Merkle trees are used in distributed systems for efficient data verification. They are efficient because they use hashes instead of full files. Hashes are ways of encoding files that are much smaller than the actual file itself. Every block header in Ethereum contains three Merkle Trees for Transactions, Receipts, and States:



Source: Merkling in Ethereum; Vitalik Buterin, Ethereum Founder

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

This makes it easy for a light client to get verifiable answers to queries, such as:

- Does this account exist?
- What is the current balance?
- Has this transaction been included in a particular block?
- Has a particular event happened in this address today?

Smart Contracts

A key concept enabled by Ethereum and other blockchain-based networks is that of *smart contracts*. These are self-executing blocks of code that multiple parties can interact with, cutting out the need for trusted middlemen. Code in a smart contract can be seen as similar to the legal clauses in a traditional paper contract, but can also achieve much more expansive functionality. Contracts can have rules, conditions, penalties for non-compliance, or can kickstart other processes. When triggered, contracts execute as originally stated at the time of deployment on the public chain, offering built-in elements of immutability and decentralization.

The smart contract is a vital tool for building on the Ethereum infrastructure. Core functionality of the Hydro blockchain layer is achieved via custom contracts, as discussed later in this paper.

Ethereum Virtual Machine

The Ethereum Virtual Machine (EVM) is the runtime environment for smart contracts on Ethereum. The EVM helps to prevent Denial of Service (DoS) attacks, ensures programs remain stateless, and enables communication that cannot be interrupted. Actions on the EVM have costs associated with them, called *gas*, which depend on the computational resources required. Every transaction has a maximum amount of gas allotted to it, known as a *gas limit*. If the gas consumed by a transaction reaches the limit, it will cease to continue processing.

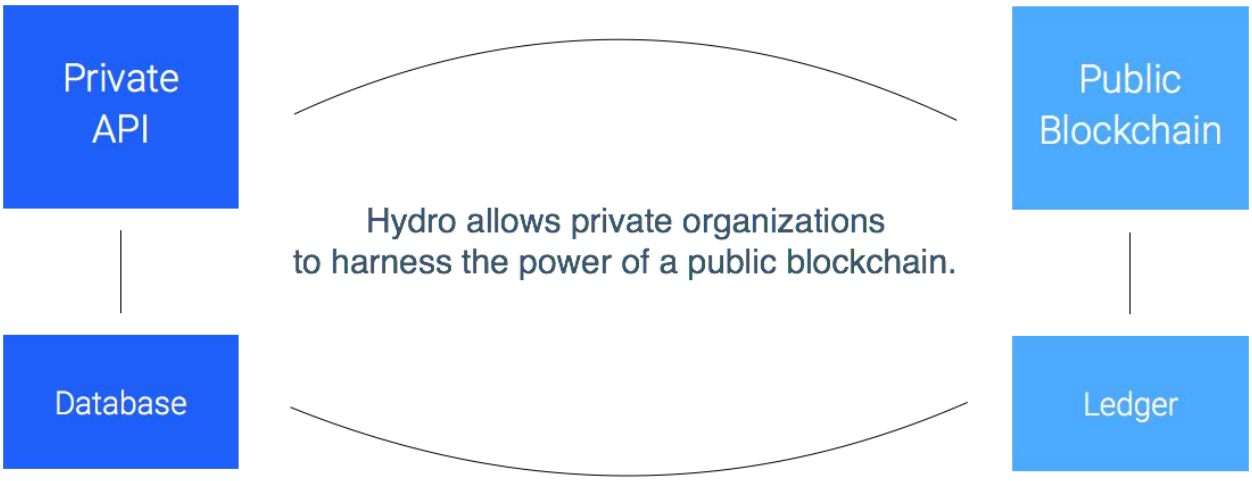
© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Public Ledger

A Public Ledger for Private Systems

The systems that power financial services platforms, websites, and applications can often be described as mediums of data flow - they send, retrieve, store, update, and process data for the entities they interface with. Because of the nature of this data, and of financial services more generally, these systems often house complex operations in a private and centralized manner. Reliance on private structures, in turn, opens the door for a variety of security, transparency, and efficiency gains to be had by incorporating external forces that exceed the reach of the internal system.

Such is the case with Hydrogen's API Platform. Hydro aims to tap into the aforementioned gains by allowing Hydrogen users to interface with a blockchain in ways that are seamlessly integrated into the fundamentally private Hydrogen ecosystem.



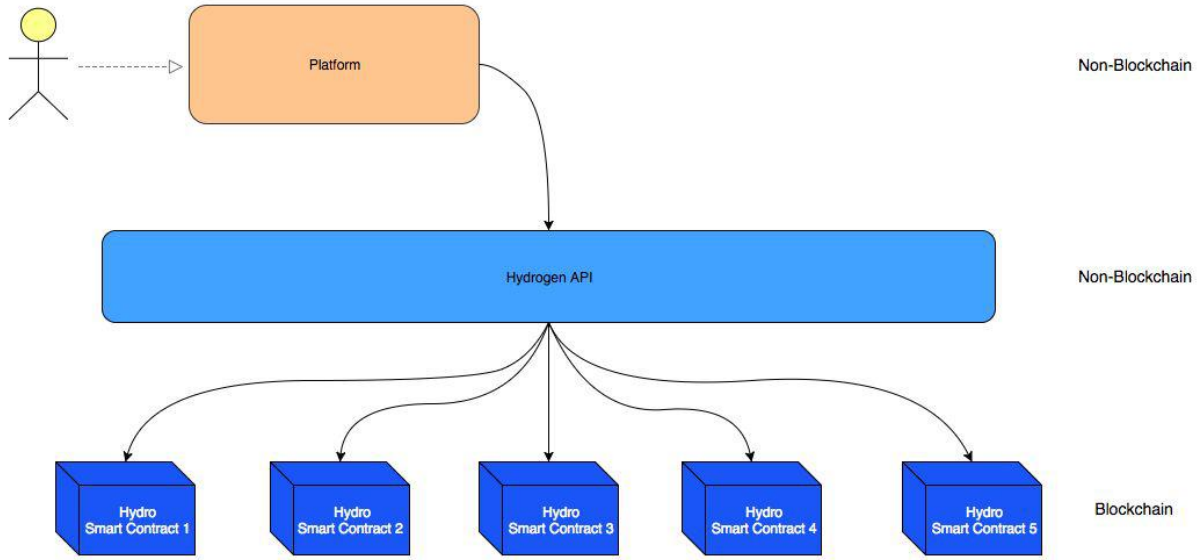
Public blockchain-based operations can occur before, during, or after private operations. The interplay between private and public elements can serve to validate, stamp, record, or enhance processes within an ecosystem.

The ethos of this model is making processes more robust by tapping into the benefits of blockchain technology specifically where it can produce the most positive impact. While this hybrid framework may not be applicable to all platforms, Hydro focuses on providing value for the cases in which it is.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Architecting for Adoption

Hydro differs from many existing blockchain initiatives, because it can exist independently and layer around new or existing systems without requiring systemic change. Rather than replace, Hydro aims to augment. Platforms and institutions that plug into the Hydrogen APIs can automatically access the blockchain.



The scope of financial services platforms that can leverage Hydrogen is broad. These platforms can power virtually any experience, house any number of proprietary services, perform any private data operation, and deploy in any environment. This is enabled by Hydrogen's structural modularity and is synergistic with Hydro, acting as a complementary driver of adoption.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Raindrop

Built on top of this Hydro public ledger is a blockchain-based authentication service, called "Raindrop." This offers a distinct, immutable, globally viewable layer of security that verifies an access request is coming from an authorized source.

Private authentication protocols such as OAuth 2.0 offer varying levels of robustness and usefulness for the spectrum of use cases that exist. There is little need to compete with or attempt to replace these protocols - Hydro offers a way to enhance them by incorporating blockchain mechanics as a component of an authentication procedure. This can add a useful layer of security to help thwart system breaches and data compromises.

Before examining technical aspects of Raindrop, let's first take a look at the problem it is trying to solve.

The State of Financial Security

The rise of the data age has brought with it a rise in vulnerability, and this is particularly important for financial services. Financial platforms are often gateways to large quantities of private and sensitive data such as government ID numbers, account credentials, and transaction histories. Because of how critically important this data is, unwarranted access is typically met with catastrophic results.

Industry research firm Trend Micro published a report that found stolen line items of Personally Identifiable Information (PII) is sold on the Deep Web for as little as $1, scans of documents like passports are available for as little as $10, and bank login credentials for as little $200, making the distribution of stolen data increasingly fragmented and untraceable.

Unfortunately, the existing financial system does not have a spotless track record when it comes to preventing, diagnosing, and communicating data breaches with its stakeholders.

- According to a recent study by Javelin Strategy & Research - *The 2017 Identity Fraud Study* - $16 billion was stolen from 15.4 million U.S. consumers in 2016 due to failures of the financial system to protect Personally Identifiable Information (PII).

- In April 2017, Symantec published its *Internet Security Threat Report*, which estimates 1.1 billion pieces of PII were compromised in various capacities over the course of 2016.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

- The *2016 Year End Data Breach Quickview* by Risk Based Security, found that 4,149 data breaches occurred in businesses globally in 2016, exposing over 4.2 billion records.

- The *2017 Thales Data Threat Report – Financial Services Edition*, a survey of global IT professionals in professional services, found that 49% of financial services organizations have suffered a security breach in the past, 78% are spending more to protect themselves, but 73% are launching new initiatives related to AI, IoT, and cloud technologies before preparing appropriate security solutions.

Equifax Breach

On July 29th 2017, Equifax, a 118 year old U.S. credit reporting agency, was hacked. 143 million consumers had PII exposed, including Social Security Numbers. 209,000 customers had credit card data compromised.

What was the cause of this breach?

It starts with one of the backend technologies utilized by Equifax. Struts is an open source framework for developing web applications in the Java programming language, built by the Apache Software Foundation. CVE-2017-9805 is a vulnerability in Apache Struts related to using the Struts REST plugin with the XStream handler to handle XML payloads. If exploited, it allows a remote unauthenticated attacker to run malicious code on the application server to either take over the machine or launch further attacks from it. This was patched by Apache two months before the Equifax breach.

Apache Struts contains a flaw in the REST Plugin XStream that is triggered as the program insecurely de-serializes user-supplied input in XML requests. More specifically, the problem occurs in XStreamHandler's toObject() method, which does not impose any restrictions on the incoming value when using XStream deserialization into an object, resulting in arbitrary code execution vulnerabilities.

Even if this REST plugin was compromised, should it have mattered? Is there a way to use blockchain technology to secure the financial information of these 143 million customers while still relying on incumbent REST API and Java-based systems?

Adding a Blockchain Layer

It is clear that the integrity of financial data gateways can be improved. Let's examine how an additional layer of security is achieved via Hydro.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

The fundamental consensus mechanisms of the Ethereum network ensure transactional validity because participants collectively process transactions that are properly signed. This reality leads to decentralization and immutability, but, more importantly, it provides a vector for mitigating unauthorized access to a gateway that handles sensitive data.

With Hydro, authentication can be predicated upon transactional operations on the blockchain. An API, for example, can choose to validate developers and applications by requiring them to initiate particular transactions, with particular data payloads, between particular addresses on the blockchain, as a precondition that kickstarts a standard authentication protocol.

The Hydro Raindrop

Rain contains packets of condensed water ranging from 0.0001 to 0.005 centimeters in diameter. In a typical rainstorm, there are billions of these packets, each of random size, velocity, and shape. Because of that, one cannot reliably predict the exact nature of rain. Similarly, every Hydro authentication transaction is unique and virtually impossible to have occurred by chance - that is why we call them *Raindrops*.

Financial services platforms commonly utilize *micro-deposit* verification to validate client accounts. The concept is simple: the platform makes small deposits of random amounts into a user's claimed bank accounts. In order to prove the user indeed owns said account, he or she must relay the deposit amounts back to the platform, which are then validated. The only way the user can know the valid amounts (besides guessing) is by accessing the bank accounts in question.

Raindrop-based verification with Hydro is analogous. Rather than sending the user an amount and having it relayed back, we define a transaction and the user must execute it from a known wallet. The only way the user can conduct a valid transaction is by accessing the wallet in question.

By using Raindrops, both the system and the accessor can monitor authorization attempts on an immutable public ledger. This blockchain-based transaction is decoupled from the basic system operations, occurs on a distributed network, and depends upon the ownership of private keys. Therefore, it serves as a useful validation vector.

A Detailed Look

There are four entities involved in the Hydro authentication process:

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

1. *Accessor* - The party attempting to access a system. In the case of Hydrogen, the accessor is a financial institution or app utilizing the Hydrogen APIs for its core digital infrastructure.
2. *System* - The system or gateway that is being accessed by the Accessor. For Hydrogen, the system is the Hydrogen API itself.
3. *Hydro* - The module that is utilized by the System to communicate and interface with the blockchain.
4. *Blockchain* - The distributed public ledger that processes HYDRO transactions and contains the Hydro smart contracts, through which information may be pushed, pulled, or otherwise operated upon.

Each Raindrop, in its entirety, is a set of five transactional parameters:

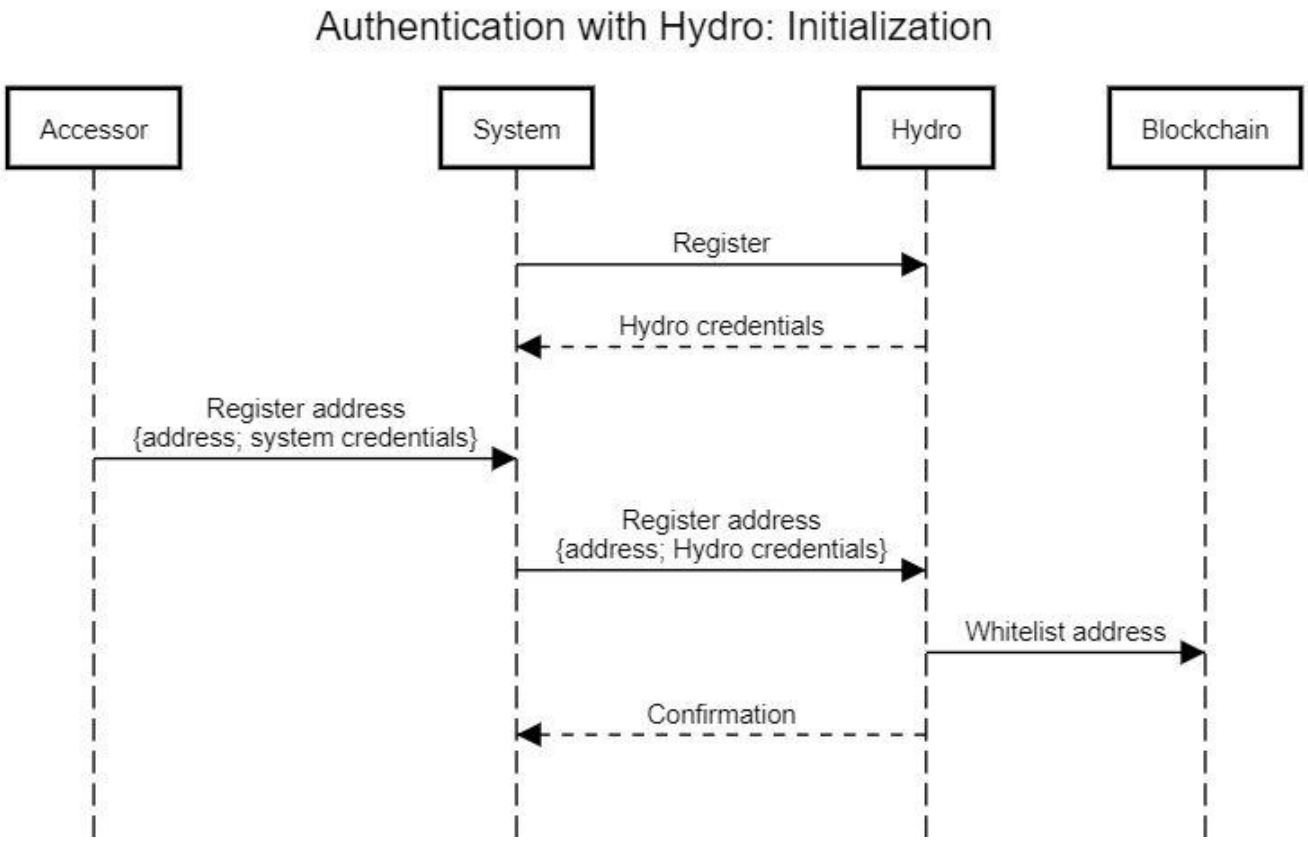
1. *Sender* - The address that must initiate the transaction.
2. *Receiver* - The transaction's destination. This corresponds to calling a method in a Hydro smart contract.
3. *ID* - An identifier that is associated with the System.
4. *Quantity* - A precise number of HYDRO to send.
5. *Challenge* - A randomly generated alphanumeric string.

Below is an outline of the authentication process, which can be generally classified into three stages:

1. Initialization
2. Raindrop
3. Validation

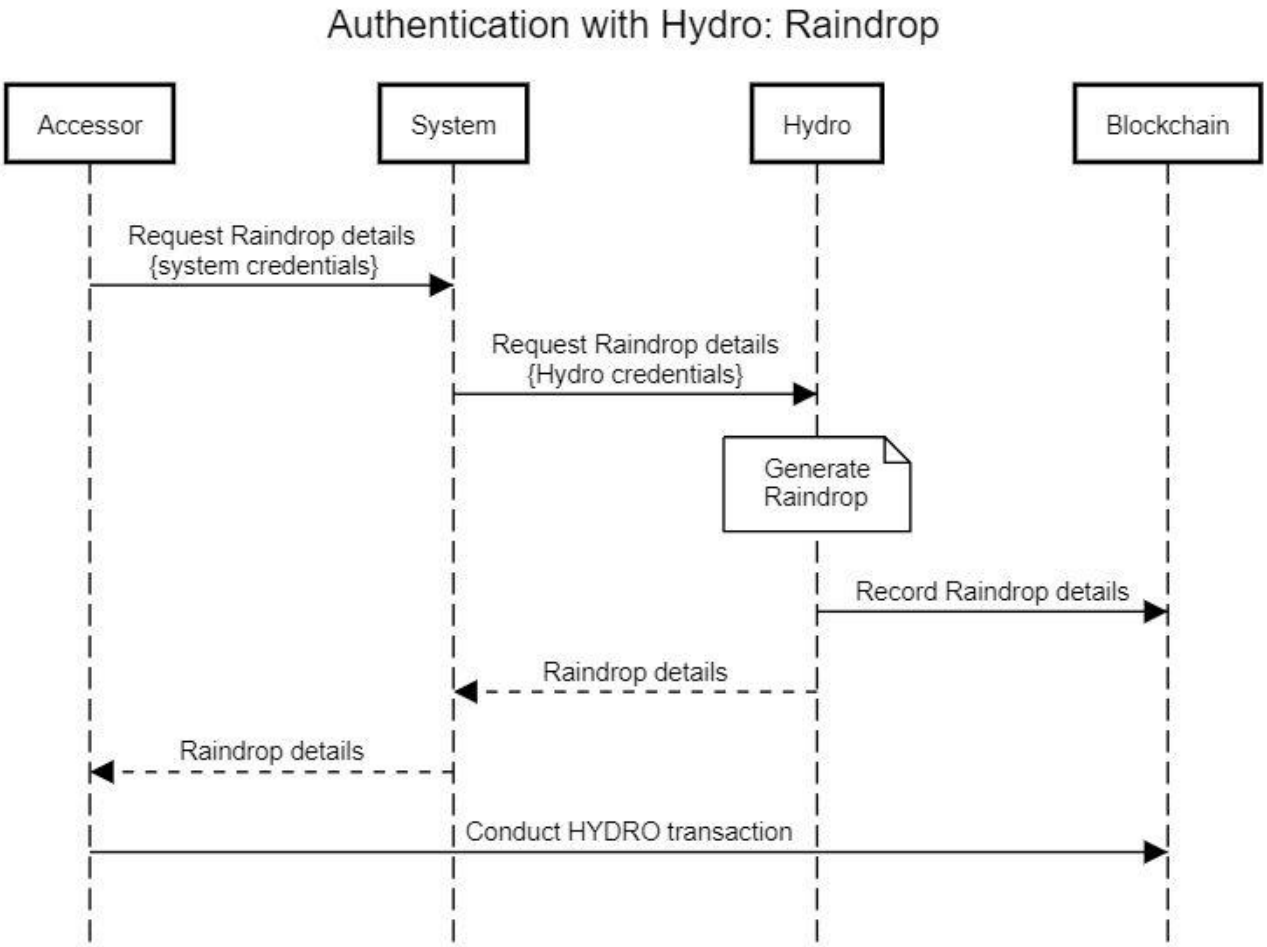
Initialization begins with a System (e.g. Hydrogen) registering to use Hydro and obtaining credentials, enabling the system to communicate with the blockchain via the Hydro module. The System onboards an Accessor (e.g. a financial institution) who registers a public address, and then passes the registered address to Hydro. This address is immutably written onto the blockchain to a whitelist stored in a Hydro smart contract. The System receives a confirmation that the address was whitelisted, which can also be verified as a publicly viewable event. System registration need only occur once, while Accessor whitelisting need only occur once per Accessor.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.



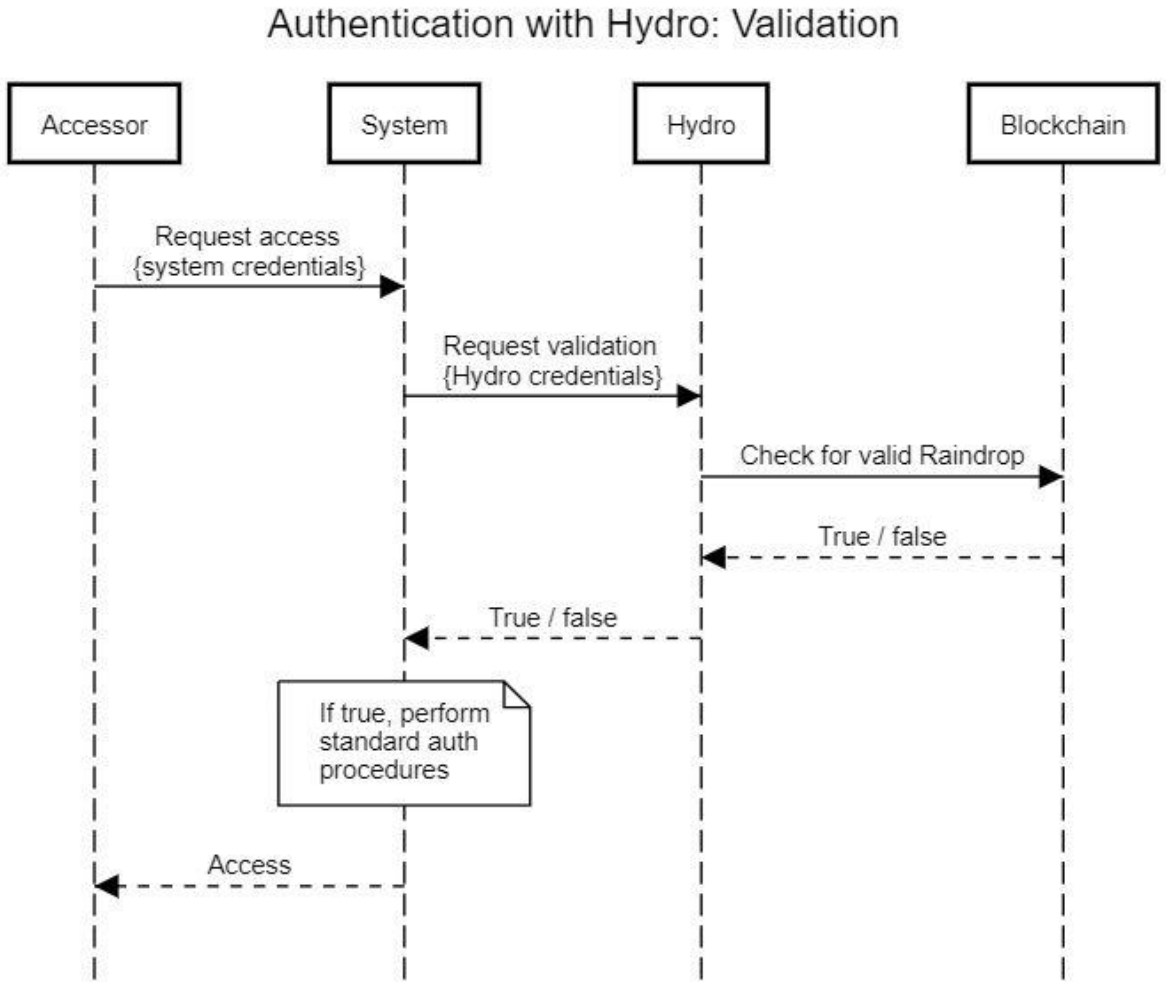
After Initialization is complete, the core of the Hydro authentication process can begin. The Accessor, who must execute a Raindrop transaction, jumpstarts this process by requesting Raindrop details from the System, and the System routes the request to Hydro. Hydro generates a new Raindrop, stores certain details immutably on the blockchain, and returns the full details to the Accessor via the System. The Accessor, equipped with all required information, conducts a transaction from the registered address to a method in the Hydro smart contract. If the address is not whitelisted, the action is rejected - otherwise, it is recorded in the smart contract. It is important to note that this transaction should occur outside of the System, directly from the Accessor to the Blockchain, as it must be signed with the Accessor's private key (which only the Accessor should be able to obtain).

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.



The final step of the process is Validation. In this step, the Accessor officially requests access to the System via the System's established mechanism. Prior to implementing any of its standard authentication protocols, the System asks Hydro whether or not the Accessor has performed a valid Raindrop transaction. Hydro interfaces with the smart contract, checks for validity, and responds with a true/false designation. The System is able to decide how it should proceed based on this designation - if it is false, the System can deny access, and if it is true, the System can grant access.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.



If we consider the base System credentials - or whatever existing System protocol that is in place - to broadly be one factor of authentication, it is important that the Hydro layer provides a useful second factor. By examining the two primary attack vectors, we can readily confirm its usefulness:

- Vector 1 - Attacker steals the Accessor's base System credentials
    - Attacker attempts to gain access to the System with valid System credentials
    - System checks with Hydro to determine if a valid transaction was made on the blockchain
    - Hydro returns false, and the System denies access
- Vector 2 - Attacker steals the private key(s) to the Accessor's wallet
    - Attacker attempts to conduct a Hydro transaction from the registered address, without required Raindrop details
    - Attacker cannot make a valid blockchain transaction

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

- Attacker also cannot request access to the System without the proper System credentials

It is clear that the Attacker must steal both the base System credentials and the Accessor's private wallet key(s) in order to access the System. In this regard, Hydro has successfully added an additional factor of authentication.

Opening The Raindrop To The Public

While this blockchain-based authentication service was architected to help secure the Hydrogen API ecosystem, it is widely applicable to different platforms and systems. Because we feel that others can potentially benefit from this verification layer, we are opening it up for use.

Just as Hydrogen will integrate it as a precondition for access to its API ecosystem, so too can any system add it to existing procedures and protocols. Any platform - be it an API, application, enterprise software, gaming platform, etc. - can leverage Hydro for authentication purposes. Formal documentation will be available on GitHub for those who wish to incorporate this blockchain layer into an authentication framework or REST API.

Case Study - Raindrop With OAuth 2.0

There are dozens of ways the Raindrop release can be used by private organizations. Private APIs, databases, and networks have created elaborate systems of tokens, keys, apps, and protocols over the last decade, in an attempt to secure sensitive data. Google, for example, became one of the most popular product providers in the market with the Google Authenticator app. As mentioned previously, there is little to no reason to compete with or replace these existing protocols.

As a case study, here is a brief overview of how Hydrogen implements Hydro authentication as a security layer in its overall API security framework:

1. Hydrogen API partners must first have the IP addresses of their various environments whitelisted.
2. Partners must request to whitelist a public Hydro address.
3. All calls to the Hydrogen APIs and transfers of data are encrypted and transmitted through the HTTPS protocol.
4. Partners must complete a valid Hydro raindrop transaction from the registered Hydro address.
5. Partners must use OAuth 2.0 validation. OAuth (Open Authorization) is an open standard for token-based authentication and authorization. Hydrogen supports the “Resource Owner Password Credentials” and “Client Credentials” grant types, and each API user must provide credentials for an authentication request.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

6. If none of the five elements above are violated, the Hydrogen partner is granted a unique token, to be checked and verified with each API call.
7. The token is valid for 24 hours, after which the partner must validate themselves again.

If any of these steps is violated, the user is immediately locked from API access. A hacker cannot bypass these security factors by guessing randomly, because there are trillions of unique combinations.

Hydro blockchain-based authentication is an important component of the Hydrogen security protocol. The Hydrogen team encourages partners to set up multi-signature wallets, and store private keys in multiple secure locations independently from other credentials, so there is not a single point of failure. A properly secured multi-signature wallet is not only difficult to steal, but the public nature of the blockchain also allows for swift recognition of any theft as it relates to the security of the API.

Anyone can view an authentication attempt to the Hydro smart contract, which means the days of platforms being compromised for months on-end can be a thing of the past. API hackers can now be thwarted with more immediacy because of the ability to detect unexpected authorization attempts in real-time, from anywhere in the world.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Risks

Much like any nascent technology, such as the early days of social media, email, and streaming applications (which were reliant on dial-up connectivity), it is important that the core development team closely track new developments in Ethereum transaction speeds and volumes. Could you imagine YouTube attempting to launch in 1995? Or Instagram being first offered on the Blackberry?

Core Ethereum developers such as Vitalik Buterin and Joseph Poon have proposed the Plasma: Scalable Autonomous Smart Contracts upgrade to the Ethereum protocol:

> **Plasma is a proposed framework for incentivized and enforced execution of smart contracts which is scalable to a significant amount of state updates per second (potentially billions) enabling the blockchain to be able to represent a significant amount of decentralized financial applications worldwide. These smart contracts are incentivized to continue operation autonomously via network transaction fees, which is ultimately reliant upon the underlying blockchain (e.g. Ethereum) to enforce transactional state transitions.**

Others, such as The Raiden Network, have proposed an off-chain scaling solution designed to power faster transactions and lower fees. At this time, the Raindrop **will put very minimal strain** on the Ethereuem framework, thus scalability is a very small risk to the success of the technology.

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.

Conclusion

The immutability of a public blockchain offers new ways to enhance security of private systems like APIs.

This paper has shown three important things:

1. Public blockchains can add value in financial services.
2. The Hydro Raindrop can enhance security of private systems.
3. There are immediate applications of the Hydro Raindrop within the Hydrogen API platform.

The Hydro team believes the framework set forth can be the standard security infrastructure for a new model of hybrid private-public systems, which will benefit all stakeholders in the financial services industry and beyond.

Sources:

Ethereum; Merkling in Ethereum
Trend Micro; What Do Hackers Do With Your Stolen Identity?
Javelin Strategy & Research; *The 2017 Identity Fraud Study*
Symantec; *Internet Security Threat Report*
Risk Based Security; *2016 Data Breach Trends - Year in Review*
Thales; *2017 Thales Data Threat Report – Financial Services Edition*
Apache.org; Apache Struts 2 Documentation - S2-052
Joseph Poon and Vitalik Buterin; Plasma: Scalable Autonomous Smart Contracts

© 2018 The Hydrogen Technology Corporation. All Rights Reserved.