Hydro Raindrop

Offentlig Verifiering På Blockkedjan

Januari 2018

INNEHÅLL

Abstract

HYDRO: Etymologi - Från Antik Grekiska ὑδρο- *hudro-*), från ὕδωρ (*húdōr*, "vatten")

Hydro möjliggör att nya och redan existerande privata system sömlöst kan integreras och influera den oföränderliga samt transparenta dynamiken hos en offentlig blockkedja för att förbättra applikations- och dokumentsäkerhet, identitetshantering, transaktioner samt artificiell intelligens.

I den här artikeln läggs det fram argument för att privata system, såsom APIs, ska använda sig av Hydros offentliga blockkedja för att förbättra säkerheten genom offentlig verifiering.

Den föreslagna teknologin kallas "Raindrop" - en transaktion som utförs genom ett smart kontrakt som bekräftar privat systemåtkomst offentligt och som kan komplementera redan existerande privata verifieringsmetoder. Teknologin är tänkt att tillföra ytterligare säkerhet för känslig finansiell data som är under ökande risk från hacking och säkerhetsöverträdelser.

Den inledande implementationen av Hydro Raindrop kommer att utföras på Hydrogen API Platform. Denna modulbaserade uppsättning av APIs är tillgänglig globalt för företag och utvecklare för att göra prototyper, bygga, testa och utplacera sofistikerade finansiella teknologi-plattformar samt produkter.
Hydro Raindrop kommer att göras tillgänglig för världens utvecklare genom öppen källkod som låter utvecklare integrera Hydro Raindrop med vilken Rest API som helst.

Blockkedjan & Ethereum

Hydro är implementerad på Ethereumnätverket. Innan mer detaljer ges om projektet så är det viktigt att förstå en del om blockkedjan samt Ethereums grundläggande principer.

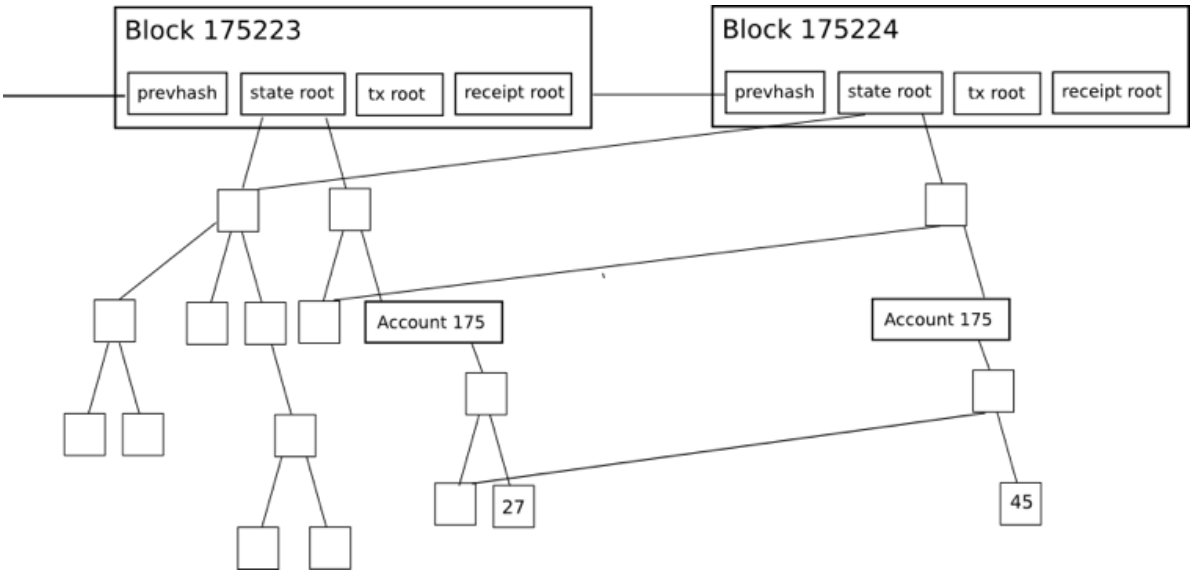
Bygga på Ethereum

Precis som att appar såsom Snapchat byggdes med Swift och andra verktyg som som tillhandahölls ovanpå Apple iOS-plattformen så kan också blockkedje-applikationer byggas ovanpå Ethereum. Snap Inc. behövde inte bygga iOS själva, utan använde den som en bas för att lansera sin applikation avsedd för sociala medier.

Project Hydro är liknande. Den förlitar sig på de tusentals utvecklare globalt som arbetar med att göra den bakomliggande blockkedjeteknologin snabbare, starkare och mer effektiv. Hydro använder sig av denna konstant förbättrade bas genom att utveckla produktfokuserade interaktioner som baseras på blockkedjeteknologi vilket kan konkret gagna finansiella serviceapplikationer.

Merkle Trees

Merkle trees används i distribuerade för en effektiv dataverifiering. De är effektiva pga att de använder hashar istället för hela filer. Hashar är ett sätt att koda filer som är mycket mindre än den faktiska filen. Varje blockhuvud i Ethereum innehåller tre Merkle Trees för Transaktioner, Kvitton samt Tillstånd:



Source: Merkling in Ethereum; Vitalik Buterin, Ethereum Founder

Det här gör att det är lätt för en light client få kontrollerbara svar på förfrågningar, såsom:

- Existerar detta konto?
- Vad är det nuvarande saldot?
- Har den här transaktionen inkluderats i ett särskilt block?
- Har en särskild händelse inträffat på den här adressen idag?

Smarta Kontrakt

Ett av Ethereuems och andra blockkejdebaserade nätverks nyckelkoncept är smarta kontrakt. Dessa är självexekverande kodblock som flera deltagare kan interagera med vilket leder till att behovet av en mellanman elimineras. Koden i ett smart kontrakt kan ses som jämförbart med de legala klausuler i ett papperskontrakt, men den kan utöver det utföra mycket mer omfattande funktioner. Kontrakten kan ha regler, förutsättningar, straff för överträdelser eller så kan de sätta igång andra processer. När de triggas så exekveras kontrakten vid tidpunkten för utplaceringen på den offentliga kedjan och erbjuder inbyggda faktorer såsom oföränderlighet samt decentralisering. Smarta kontrakt är ett vitalt verktyg vid påbyggnad av Ethereums bas. Kärnfunktionerna i Hydros blockkedjelager uppnås via skräddarsydda kontrakt, vilket kommer att beröras senare i denna artikel.

Ethereum Virtual Machine

The Ethereum Virtual Machine (EVM) är körmiljön för smarta kontrakt på Ethereum. EVM hjälper till att förhindra Denial of Service (DoS) attacker, ser till att program fortsätter vara "stateless", och möjliggör kommunikation som inte kan avbrytas. Handlingar på EVM har kostnader knutna till dem, som kallas "gas", som beror på vilken resurs som behövs.

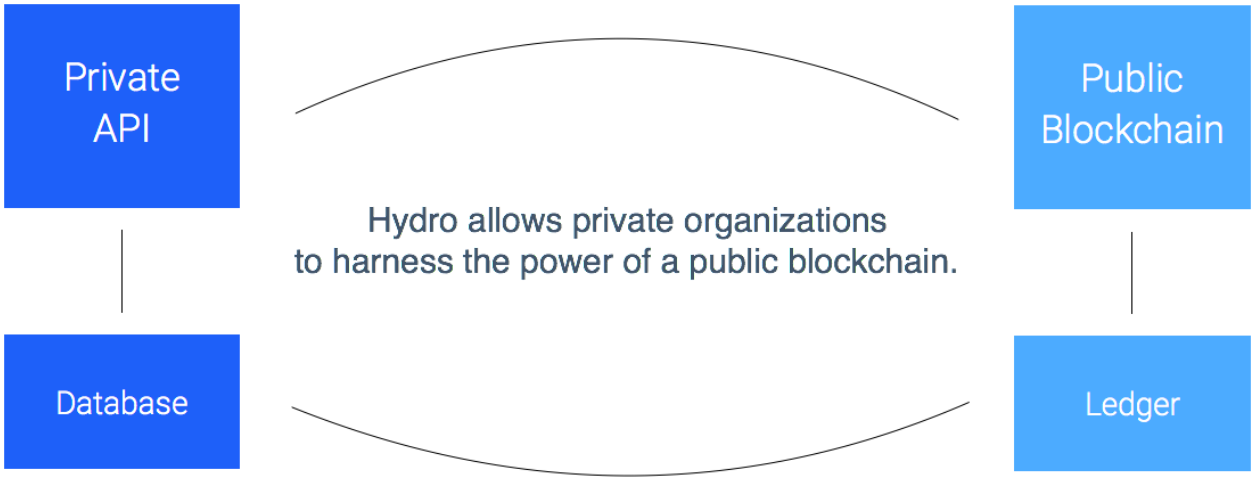
Varje transaktion tilldelas en maximal mängd av gas, som kallas "gas limit". Om gasen som konsumeras av en transaktion når sin begränsning så slutar den att utföra processen.

Public Ledger

En Public Ledger för Privata System

Systemen som driver finanstjänst-plattformar, webbsidor och applikationer kan beskrivas som dataflöde-medier - de sänder, tar emot, lagrar, uppdaterar och processerar information. Pga den här datans natur, och det gäller finanstjänster mer generellt, så hyser dessa system ofta komplexa funktioner på ett privat och centraliserat sätt. Att förlita sig på privata strukturer öppnar dörren för en mängd fördelar inom säkerhet, transparens och effektivitet då man införlivar externa krafter som överträffar det interna systemets räckvidd.

Så är fallet med Hydrogens API Plattform. Hydro strävar efter att exploatera dessa fördelar genom att tillåta Hydrogen-användare att interagera med blockkedjan på sätt som är sömlöst integrerade med det i grund och botten privata Hydrogen ekosystemet.

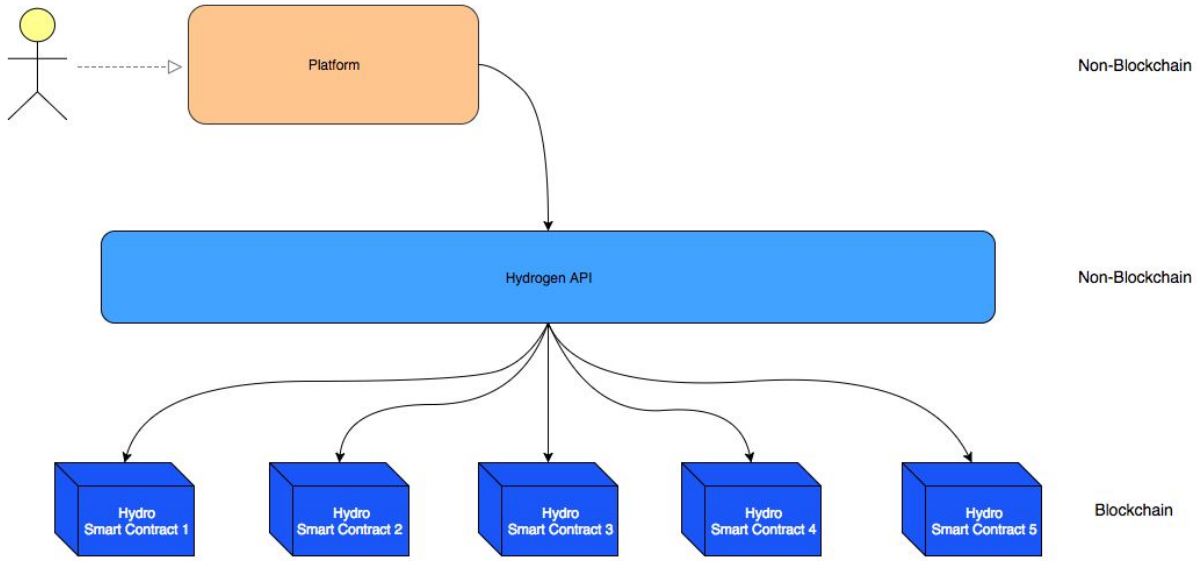


Offentliga blockkedje-baserade funktioner kan ske innan, under eller efter privata funktioner. Samspelet mellan privata och offentliga element kan tjäna till att bekräfta, stämpla, registrera eller förbättra processerna inom ett ekosystem.

Modellens grundtanke är att göra processerna mer robusta genom att använda sig av blockkedje-teknologins fördelar där de kan ge mest positiv effekt. Dock så kan denna sorts hybridstruktur inte vara tillämplig på alla sorters plattformar, Hydro fokuserar på att generera värde i de fall där den är tillämplig.

Konstruera för Adoptering

Hydro skiljer sig åt från många redan existerande blockkedje-initiativ genom att den kan existera självständigt och lägga sig som ett skikt runt nya eller existerande system utan att kräva ett systembyte. Istället för att ersätta något så riktar Hydro in sig på att förstärka. Plattformar som ansluter till Hydrogen APIs får automatiskt åtkomst till blockkedjan.



Spännvidden på de finanstjänst-plattformar som kan använda Hydrogen är bred. Dessa plattformar kan driva i princip vilken händelse som helst, ha plats för vilken mängd av äganderättstjänster som helst, utföra vilken privat datafunktion som helst samt placeras ut i vilken miljö som helst. Detta möjliggörs av Hydrogens strukturella modularitet och är synergistisk med Hydro som agerar som en kompletterande frammanare av adoptering.

Raindrop

Ovanpå Hydro Public Ledger så har det byggts en blockkedje-baserad verifieringstjänst, som kallas "Raindrop". Raindrop erbjuder ett distinkt, oföränderligt, globalt visningsbart säkerhetslager som verifierar att en åtkomstbegäran verkligen kommer från en auktoriserad källa.

Privata verifieringsprotokoll, såsom OAuth 2.0, erbjuder en varierande nivå av robusthet och användningsbarhet för det stora spektrum av ärenden som förekommer. Det finns ingen större mening med att tävla med dessa protokoll - Hydro erbjuder istället ett sätt för att förbättra dessa genom att införliva dem med blockkedje-mekanismer som en komponent i åtkomstproceduren. Detta lägger till ett användbart säkerhetslager som kan hjälpa till att stävja systeminbrott.

Innan vi belyser Raindrops tekniska aspekter så tar vi först en titt på det problem som Raindrop försöker lösa.

Den Finansiella Säkerhetens Tillstånd

Dataåldern har fört med sig en ökande sårbarhet och detta är särskilt viktigt att stävja för finanstjänster. Finansplattformar är ofta ingångsportar till stora kvantiteter av privat och känslig data såsom personnummer, kontouppgifter samt transaktionshistorik. P.g.a. hur viktig denna data är så leder ofta otillbörlig tillgång till katastrofala konsekvenser.

Branschforskarna på Trend Micro publicerade en rapport som fann att Personally Identifiable Information (PII) säljs på Deep Web för så lite som $1, inscanningar av pass säljs för $10, och bankinloggningsuppgifter säljs för så lite som $200, vilket gör distributionen av stulen data väldigt fragmenterad och ospårbar.

Tyvärr så har inte det existerande finanssystemet några fläckfria meriter avseende att stoppa, diagnosera och kommunicera datainbrott till sina användare.

- Enligt en nygjord studie av Javelin Strategy & Research - *The 2017 Identity Fraud Study* - stals $16 miljarder från 15.4 miljoner Amerikaner under 2016 p.g.a. misslyckade försök av finanssystemet att skydda Personally Identifiable Information (PII).

- I April 2017 publicerade Symantec sin *Internet Security Threat Report*, som uppskattade att 1.1 miljarder delar av PII kompromitterades i olika omfattning under 2016 års gång.

- *2016 Year End Data Breach Quickview* av Risk Based Security, fann att 4,149 datainbrott utfördes globalt mot företag under 2016, vilket exponerade över 4.2 miljarder dokument.

- *2017 Thales Data Threat Report – Financial Services Edition*, en undersökning baserad på svar från professionella IT-tjänstemän, fann att 49% av de organisationer som tillhandahåller finanstjänster har utsatts för säkerhetsöverträdelser, 78% spenderar mer för att skydda sig själva, men 73% lanserar nya initiativ inom AI, IoT och molnteknologi innan de har förberett sig med säkerhetslösningar.

Equifax Breach

Den 27:e juli 2017 blev det 118 år gamla kreditupplysningsföretaget Equifax hackat. 143 miljoner konsumenter fick sin PII exponerat, inkl. deras personnummer. 209000 kunder fick sin kreditkort kompromitterade.

Vad var anledningen till detta dataintrång?

Det började med en av backend-teknologierna som användes av Equifax. Struts är ett open source-ramverkför att utveckla webb-applikationer i Java som är ett programmeringspråk, av Apache Software Foundation. CVE-2017-9805 är en sårbarhet i Apache Struts som härrör med att man använder Struts REST plugin med XStream handeler för att hantera XML-nyttolaster. Om detta utnyttjas så kan en overifierad anfallare att köra skadlig kod på servern för att antingen ta över maskinen eller iscensätta attacker från den. Apache patchade detta två månader innan the Equifax breach.

Apache Struts innehåller en brist i REST Plugin XStream som utlöses av att programmet avserialiserar användarens inmatning i XML förfrågningar. Mer specifikt så uppstår problemet i XStreamHandler's toObject() metod, som inte inför några restriktioner på det inkommande värdet när XStreams avserialisation används på ett objekt, vilket leder till sårbarhet genom godtycklig kodexekvering.

Även om REST plugin blev kompromitterad, spelade det någon roll egentligen? Finns det ett sätt att använda blockkedje-teknologin för att säkerställa dessa 143 miljoner kunders finansiella information, men ändå förlita sig på befintliga REST API och Java-baserade system?

Lägga till ett blockkedjelager

Det står klart att integriteten hos ingångsportarna för finansiell data kan förbättras. Låt oss undersöka hur ett extra säkerhetslager kan uppnås genom Hydro.

Det finns en grundläggande principiell mekanism hos Ethereum-nätverket som ser till att transaktionen valideras pga att deltagarna kollektivt processerar transaktioner som är signerade på ett korrekt sätt. Detta leder till decentralisering och oföränderlighet, samt att det ger en vektor för att stoppa obehörig tillgång till en inkörsport som hanterar känslig data.

Med Hydro så kan verifieringen baseras på transaktionshantering via blockkedjan. En API kan t.ex. välja att validera utvecklare och applikationer genom att kräva att de sätter igång specifika transaktioner, med särskilda nyttolaster mellan särskilda adresser på blockkedjan som en förutsättning för att kickstarta standard verifieringsprotokoll.

The Hydro Raindrop

Regn innehåller paket av kondenserat vatten som mäter 0.0001 till 0.005 cm i diameter. I typiskt regnväder så finns det miljarder sådana paket, alla har en slumpmässig storlek, hastighet och form. Pga detta så kan man inte förutspå regnets exakta natur. På ett liknande sätt så är varje Hydro verifieringstransaktion unik och det är i princip omöjligt att den har uppstått av en slump, därför kallar vi dem *Raindrops.*

Finanstjänsteplattformar använder vanligen *mikroinsättningsverifikationer* för att validera klientkonton. Konceptet är enkelt: plattformen gör små slumpmässiga insättningar till en användares bankkonto. För att kunna bevisa att användaren verkligen äger kontot så måste han In order to prove the user indeed owns said account, he vidarebefordra den insatta summan tillbaka till plattformen, som sedan är validerad. Enda sättet för användaren att veta den giltiga summan är genom att användaren har tillgång till bankkontot.

Raindrop-baserade verikfikationer med Hydro är likartade. Istället för att skicka en summa till användaren och få den återinsatt, så definierar vi en transaktion och användaren måste utföra den från en bekant plånbok. Det enda sättet för en användare att utföra en giltig transaktion är genom att ha tillgång till plånboken i fråga.

Genom att använda Raindrops, så kan både systemet och användarenboth the system and the accessor can övervaka behörighetsförsök till en oföränderlig public ledger. Denna blockkedjebaserade transaktion är frånkopplad från de basala systemoperationerna, utförs på ett distribuerat nätverk, och förlitar sig på privata nycklar. Pga detta så är det en bra valideringsvektor.

A Detaljerad Beskrivning

Det är 4 enheter som är involverade i Hydros verifikationsprocess:

1. *Accessor* - Den som försöker få access till ett system. I fallet med Hydrogen, så är accessorn ett finansinstitut eller en app som använder Hydrogen APIs för sin digitala infrastruktur.
2. *System* - Systemet som Accesor får tillgång till. För Hydrogen, så är systemet själva Hydrogen API.
3. *Hydro* - Modulen som användas av systemet för att kommunicera och interagera med blockkedjan.
4. *Blockchain* - Den distribuerade public ledger som behandlar HYDRO transaktioner och innehåller Hydro smarta kontrakt, vilken genom information kan bli ändrad.

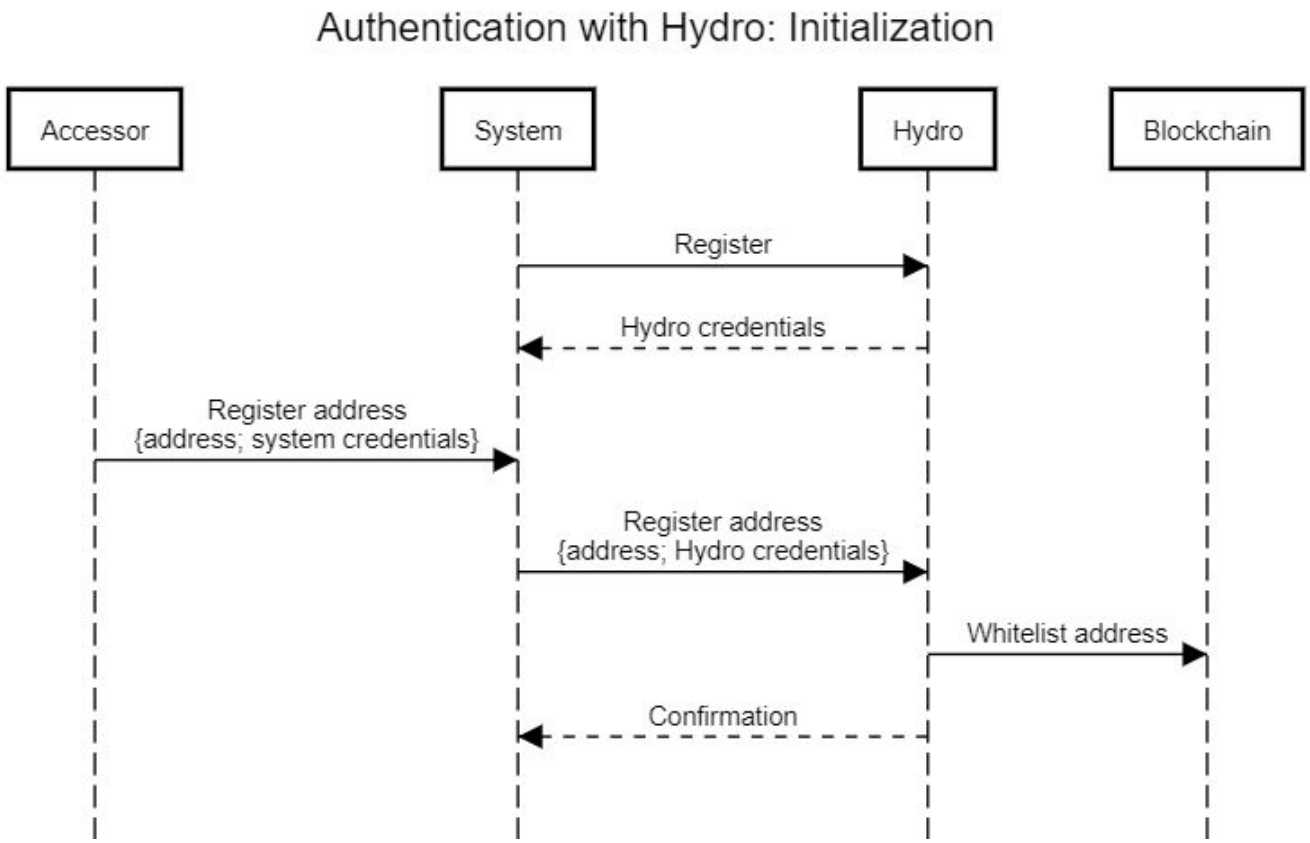
Varje Raindrop, i sin helhet, består av fem transaktionsparametrar:

1. *Sender* - Adressen som initierar transaktionen.
2. *Receiver* - Transaktionens slutmål. Detta motsvarar att kontakta en metod i ett Hydro smart kontrakt.
3. *ID* - En identifierare som associerad med Systemet.
4. *Quantity* - Det exakta antal HYDRO som skickas.
5. *Challenge* - En slumpmässigt utvald alfanumerisk sträng.

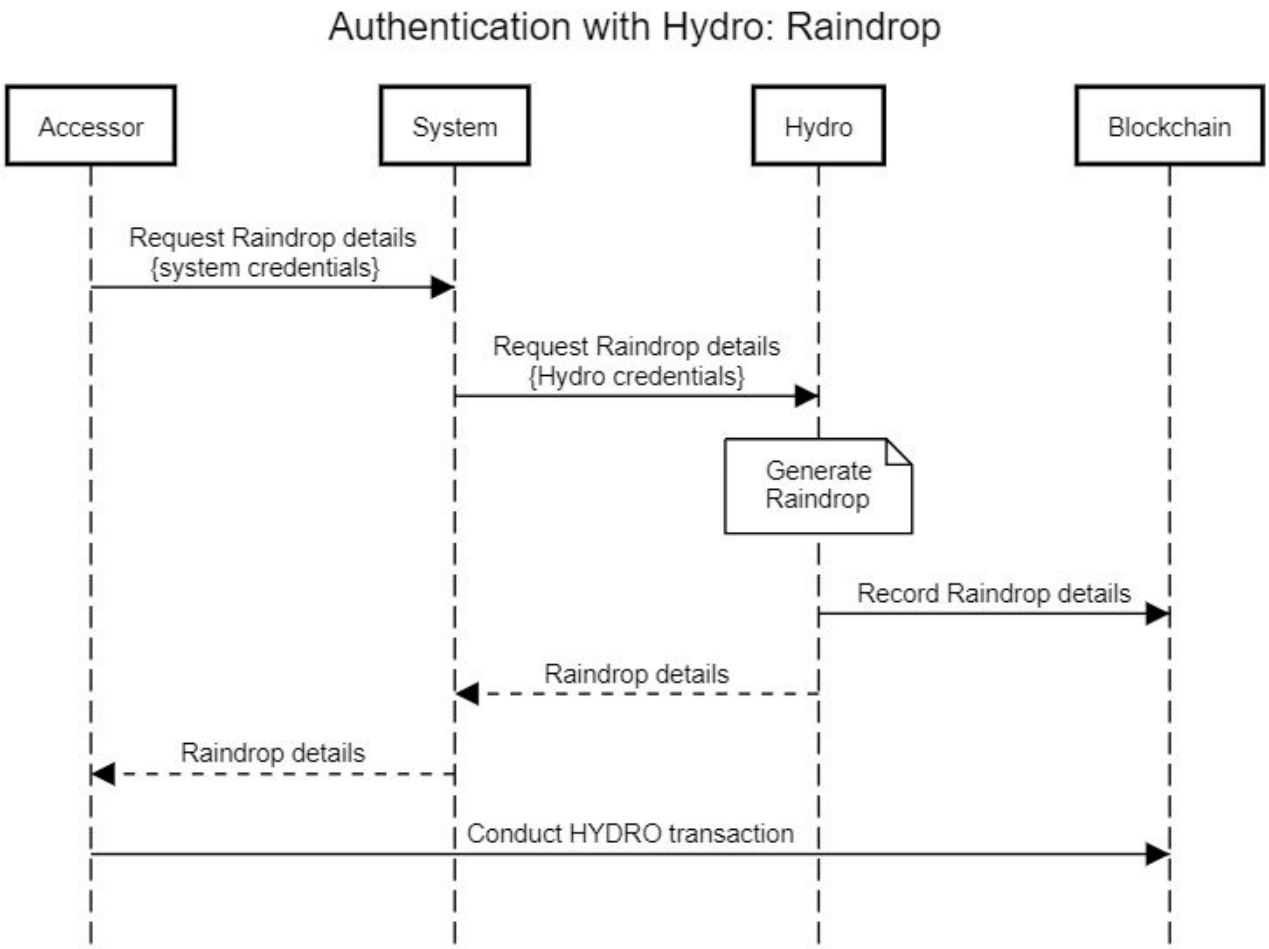
Nedan följer en redogörelse över verifikationsprocessen, som kan delas in i tre olika skeden:

1. Initialisering
2. Raindrop
3. Validering

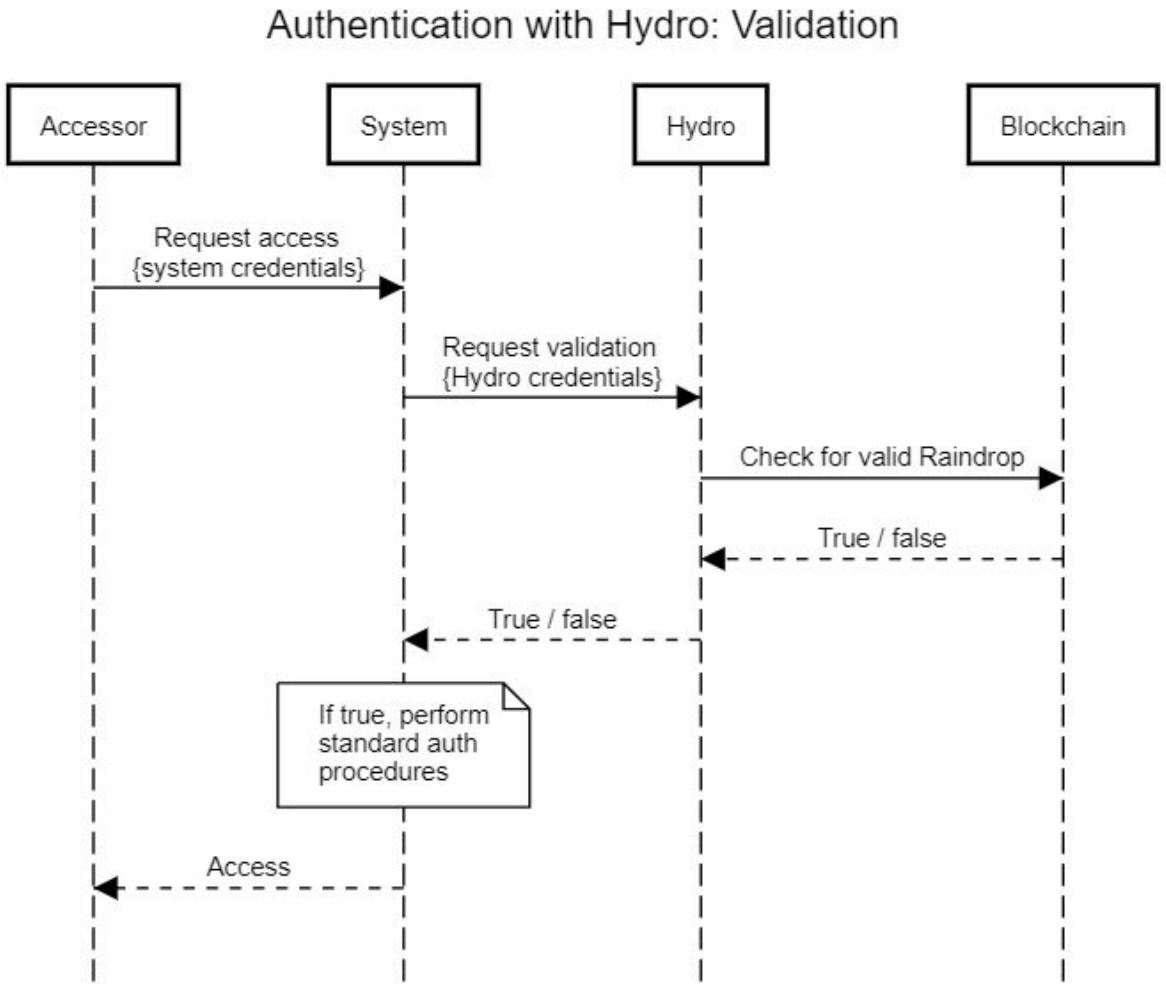
Initialiseringen börjar med ett system (t.ex. Hydrogen) som registrerar sig för att använda Hydro och som erhåller kreditiv vilket låter systemet kommunicera med blockkedjan via Hydro-modulen. Systemet tar emot en användare (t.ex. ett finansinstitut) som registrerar en offentlig adress, och skickar sedan vidare den registrerade adressen till Hydro. Denna adress skrivs oföränderligt på blockkedjan till en whitelist som lagras i ett Hydro smart kontrakt. Systemet erhåller en bekräftelse om att adressen las till i whitelistan, som också kan verifieras genom en offentligt publicerad händelse. Registreringen behöver endast göras en gång, medan användarens whitelistning endast behöver göras en gång per användare.



Efter att initialiseringen har slutförts så kan själva kärnan i Hydros verifikationsprocess inledas. Användaren, som måste utföra en Raindrop-transaktion, inleder denna process genom att begära Raindrop-detaljer från systemet och systemet i sin tur vidarebefordrar denna begäran till Hydro. Hydro genererar en ny Raindrop, lagrar vissa uppgifter oföränderligt på blockkedjan och skickar tillbaka alla uppgifter till användaren via systemet. Användaren, nu utrustad med all tillbörlig information, handhaver en transaktion från den registrerade adressen till en metod i Hydro smart kontrakt. Om adressen inte är med på whitelisten så blir åtgärden avvisad, i annat fall så skrivs den in i det smarta kontraktet. Det är viktigt att notera att denna transaktion bör utföras utanför systemet, direkt från användaren till blockkedjan, då den måste bli signerad med användarens privata nyckel (som endast användaren bör ha tillgång till).



Det sista steget i processen är validering. I det här steget så ber användaren officiellt om tillgång till systemet. Innan systemet implementerar någon av dess standard verifikationsprotokoll, så frågar Hydro systemet huruvida användaren har utfört en giltig Raindrop-transaktion. Hydro interagerar med det smarta kontraktet, Hydro interfaces with the smart contract, kontrollerar validiteten, och svarar med en sant/falskt beteckning. Systemet kan avgöra hur den ska fortsätta baserat på denna beteckning - om det är falskt så kan systemet neka tillgång och om det är sant så kan systemet bevilja tillgång.



Om vi beaktar de basala system-kreditiven - eller vilket existerande system-protokoll som finns på plats - som i stort sett en verifikationsfaktor, så är det viktigt att Hydro erbjuder en användbar andra faktor. Genom att undersöka de två primära attack-vektorerna så kan vi fastställa dess användbarhet:

- Vektor 1 - En angripare stjäl de basala system-kreditiven
  - Angriparen försöker få tillgång till systemet genom att använda giltiga system-kreditiv
  - Systemet stämmer av med Hydro för att fastställa om en giltig transaktion gjordes på blockkedjan
  - Hydro returnar falskt, och systemet avslår tillgången
- Vektor 2 - Angriparen stjäl privata nycklar till plånboken
  - Angriparen försöker genomföra en Hydro-transaktion från en registrerad adress, utan nödvändiga Raindrop-detaljer
  - Angriparen kan inte göra en giltig blockkedje-transaktion

- Angriparen kan inte heller begära tillgång till systemet utan de rätta system-kreditiven

Det står klart att angriparen måste stjäla båda de basala system-kreditiven samt användarens plånboks-nycklar för att få tillgång till systemet. På detta sätt har Hydro lagt till ännu en verifikations-faktorn.

Att Öppna Upp The Raindrop För Allmänheten

Medan denna blockkedje-baserade verifikations-service skapades för att hjälpa till att säkra Hydrogen APIs ekosystem, så kan den användas brett med ett antal olika plattformar och system. Då vi känner att andra kan dra nytta från detta verifikationslager, så öppnar vi upp det för användning.

Precis som att Hydrogen kan integreras som en förutsättning för tillgång till dess API ekosystem, så kan system lägga till den till existerande procedurer och protokoll. Varje plattform - API, applikation, företagsmjukvara etc. - kan använda Hydro för verifikationssyften. Formell dokumentation kommer att vara tillgänglig på GitHub för de som vill inkorporera detta blockkedjelager till en verifikationsstruktur eller REST API.

Fallstudie - Raindrop Med OAuth 2.0

Det finns dussintals sätt som Raindrop kan användas av privata organisationer. Privata APIs, databaser, och nätverk som har skapat genomarbetade systems av tokens, nycklar, appar, och protokoll det senaste årtiondet, i försök att säkra känslig data. Google, t.ex., blev en av de mest populära produktskaparna i segmentet med Google Authenticator appen. Som tidigare nämnts så finns ingen anledning att konkurrera med eller ersätta dessa redan existerande protokoll.

Som en fallstudie kommer här en kort översikt av hur Hydrogen implementerar Hydro verifikation som ett säkerhetslager för dess API säkerhetsstruktur:

1. Hydrogen API partners måste först och främst ha sina olika IP-adresser whitelistade.
2. Partners måste begära att whitelista en offentlig Hydro-adress.
3. Alla anrop till Hydrogen APIs och överföringar av data är krypterad och sänds genom HTTPS protokollet.
4. Partners måste slutföra en giltig Hydro raindrop transaktion från den registrerade Hydro-adress.
5. Partners måste använda OAuth 2.0 validering. OAuth (Open Authorization) är en öppen standard för token-baserad verifiering och befogenheter. Hydrogen stödjer “Resource Owner Password Credentials” och “Client Credentials” grant-typer, och varje API-användare måste tillhandahålla kreditiv för en verifikationsbegäran.

6. Om ingen av de fem första punkterna överträds så får Hydrogen partnern en unik token som granskas och verifieras vid varje API-anropning.
7. Denna token är giltig i 24 timmar, sedan måste partnern validera sig igen.

Om någon av dessa steg överträds så låses användaren omedelbart ut från APIn. En hacker kan inte kringgå de här säkerhetsfaktorerna genom att gissa, pga att det finns triljoner unika kombinationer.

Hydros blockkedjebaserade verifiering är en viktig komponent i Hydrogens säkerhetsprotokoll. Hydrogen-teamet uppmuntrar partners att upprätta multi-signatur plånböcker och att förvara sina nycklar på flera säkra platser, oberoende från andra kreditiv, så att det inte finns någon svag punkt. En ordenligt säkrad multi-signatur plånbok är inte bara svår att stjäla, utan blockkedjans offentlighet erbjuder även snabb upptäckt av stölder då det är kopplat till APIns säkerhet.

Vem som helst kan se ett verifikationsförsök till Hydros smarta kontrakt, vilket betyder att tiden då plattformar kunde bli kompromitterad i flera månader är över. API hackers kan nu stoppas snabbare pga förmågan att upptäcka oväntade verifikationsförsök i realtid från var som helst i världen.

Risker

Precis som annan begynnande teknologi, såsom social media, email och streaming-applikationer i början, så är det viktigt att kärngruppen av utvecklare noga följer utvecklingen inom Ethereums transaktionshastighet samt dess volym. Kan du tänka dig om Youtube hade försökt lansera sin produkt 1995? Eller att Instagram först började användas genom Blackberry?

Viktiga Ethereum-utvecklare såsom Vitalik Buterin och Joseph Poon har föreslagit att Plasma: Scalable Autonomous Smart Contracts används som uppgradering till Ethereum-protokollet:

> **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.**

Andra, såsom The Raiden Network, har föreslagit en off-chain scaling lösning som är designad att ge snabbare transaktioner och lägre avgifter.Just nu, så utsätter Raindrop Ethereum-ramverket för väldigt liten påfrestning, så skalbarhet utgör en väldigt liten risk för att äventyra teknologins framgång.

Slutsats

Oföränderligheten hos en offentlig blockkedja erbjuder nya sätt att förbättra säkerheten hos privata system såsom APIs

Den här artikeln har påvisat tre viktiga saker:

1. Offentliga blockkedjor kan ge mervärde inom finanstjänster.
2. Hydro Raindrop kan förbättra säkerheten hos privata system.
3. Det finns omedelbara användningsområden för Hydro Raindrop inom Hydrogen API plattformen.

Hydro-teamet tror att strukturen som har tagits fram kan bli en standard inom säkerhetsinfrastrukturen för en ny modell av system som är en hybrid mellan privata och offentliga system, vilket kommer att gagna alla intressenter inom finanstjänst-industrin.

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