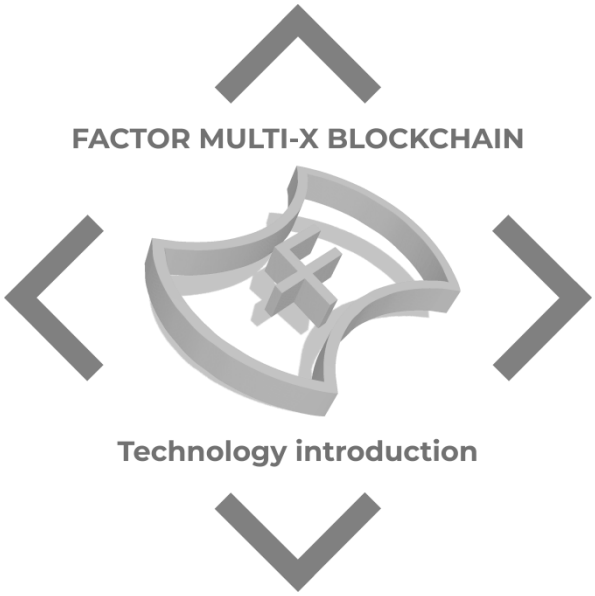
Factor Whitepaper



Version 0.1

Date updated: 06/05/2019

Factor Multi-X Blockchain

Company

서 론

과거로부터 지금까지 '기술'은 다양하게 발전해왔다. 우리가 지금 흔히 이용하는 인터넷 역시 군사기술로 시작하여 비즈니스적으로 활용되어 오다 일반인들이 이용하기 시작하면서 급격하게 확산되었다. 기술은 많은 것들을 가능하게 만드는 원동력이지만, 그것이 많은 사람들의 삶에 실제 영향을 끼치는 것은 그 기술이 활용되어 사람들에게 퍼져가는 순간부터이다. 이러한 맥락에서 블록체인도 마찬가지다. 현재 블록체인에 많은 관심과 열의를 가지고 접하는 계층은 다양하지만 대다수가 시대를 앞서 고민하며 나아가려는 사람들로 여겨진다. 중앙 집권적 기관을 배제하고 참여자 모두에게 권한을 부여하는 개인 주권주의 철학원리, 이를 가능하게 하는 IT 기술, 참여자들에 대한 기여와 보상체계 그리고 이를 활용한 비즈니스적 요소 이 모두가 블록체인에 의해 전개되는 새로운 패러다임 인 것은 분명하다.

하지만, 이처럼 블록체인이 우리의 삶에 미치는 영향력과 변화 가능성에 초점을 두고 지금의 현상들을 관찰해보면, 현재 블록체인의 발전 단계는 여전히 '초기'라는 사실을 인정할 수 밖에 없다. 지속적인 블록체인 기술의 발전에도 불구하고 새로운 블록체인 개발자들은 여전히 기존의 기술에 의존하는 방식을 사용하면서 많은 문제점과 비용소모 및 여러 가지 문제점에 직면하고 있기 때문이다. 또한 각각의 블록체인에서 풀어야 할 기술적인 문제들도 여전히 많이 존재하며 많은 블록체인들이 아이디어나 개념 수준에서 제안 개발되고 있는 현실이다.

Factor 블록체인은 이러한 문제를 해결하기 위해 개발 및 연구를 진행하였으며, 기존의 **오라클 문제 및 TPS 속도, 블록체인의 확장성과 같은 과제를 해결**하고, 향후 다가올 차세대 네트워크에 응용 가능한 블록체인을 개발할 수 있도록 할 것이다. **Factor 블록체인**은 기존의 블록체인 체제와의 **통합을 목적**에 두고 **개발**을 **진행**하고 있으며, 이것은 각각의 코인들과의 연결성 목적에 국한되기보다는, 기존 다른 블록체인들과 Factor 블록체인의 통합에 의의를 두고 있다고 할 수 있다. 기존의 블록체인들이 하나의 블록체인으로 거듭 난다면 현재 블록체인이 가지고 있는 오라클 문제나 TPS 속도, 확장성, 호환성 및 실제 생태계에서 응용했을 때 발생할 수 있는 다양한 문제들을 해결할 수 있는 돌파구가 될 것이다.

블록체인을 응용한 기술의 사용은 단순히 금융에만 활용되는 것이 아니라, 다가올 **4 차산업혁명**이 가져올 산업 생태계에서 높은 속도와 호환성 및 확장성 그리고 무한한 응용 가치를 발휘하는 혁신적인 용도로 활용될 수 있도록 하는 것이 Factor 블록체인의 궁극적인 목표이다. 나아가, 스마트폰, 가전, 자동차, 스마트홈, 기상관측, 무역 등 다양한 분야에서 경제생태계 근간을 바꿀 수 있는 혁신적인 계기가 될 것이라고 우리 Factor 블록체인 기술팀은 확신하고 있다.

Factor MX 블록체인은 기존의 블록체인들을 연결할 뿐만 아니라, 현존하고 있는 **기술적 어려움, 과도한 비용, 프로그래밍 개발언어의 불일치** 및 기존 프로그램들과 호환 및 사용이 어려운 점 등을 해결하려는 노력에만 그치지 않고, 나아가 차세대 네트워크 발전에 앞장설 것이며 블록체인의 생활화를 위해 지속적인 개발과 노력을 멈추지 않을 것이다. Factor MX 블록체인은 블록체인에 의한 다채로운 적용 가능성을 바탕으로, 신뢰에 대한 다양한 협업 시나리오를 지원할 것이며, 새로운 시나리오와 어플리케이션의 요구사항에 따라 필요한 모듈과 프로토콜들을 지속해서 늘려나갈 것이다.

블록체인은 **Factor MX 블록체인**을 통해 새롭게 정의되며, 블록체인과 우리가 접하게 될 **미래는 Factor MX 블록체인**에 의해 **실현 될 것**이다.

Index

1. 구조 및 설계

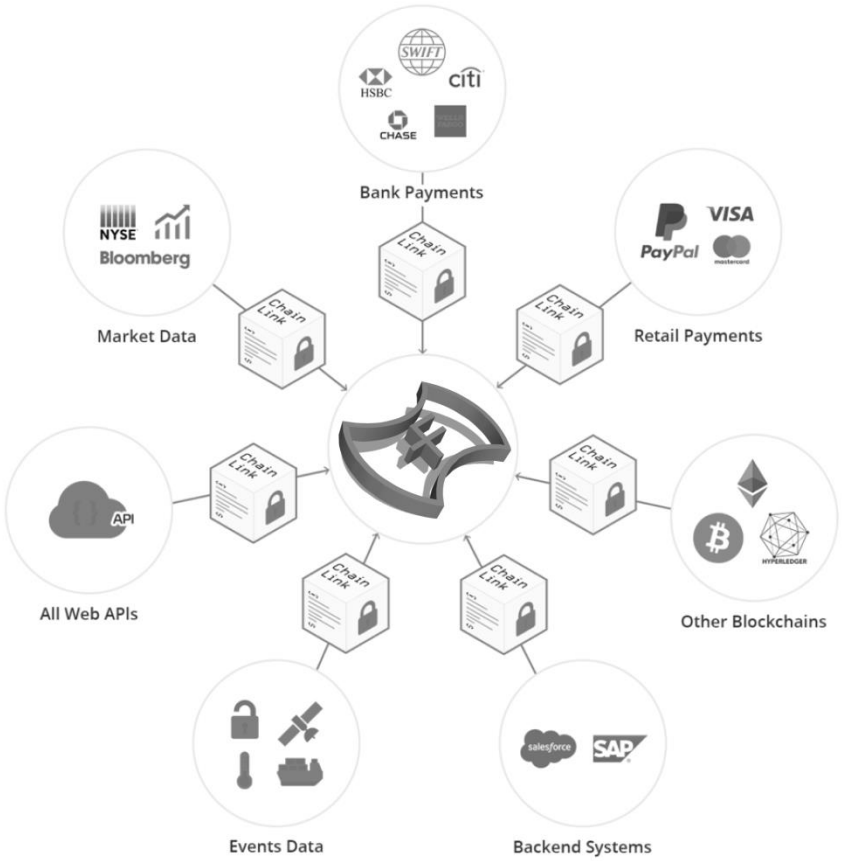
1 - 1 프레임워크 설계

Factor 블록체인은 MX (Multi-X) 블록체인을 응용하여 다중시스템 프레임 분산 원장시스템을 지원합니다. 이것은 MX 블록체인이 가진 26 개 이상의 팩터 특허 해시를 통해 속도, 확장성, 호환성을 증가시켜 주며 각 연결해시를 통해 한 블록 내에 모두 연결시켰기 때문에 모든 블록체인들과 호환이 가능하며 확장성이 우수하여 다양한 기능을 제공합니다. 이것을 통해 여러 가지 플랫폼에 사용할 수 있는 프로그램을 응용하여 사용할 수 있고 MXnode, Seed node, Masternode, Nomal node, Pos 의 스프레드 연결방식, Pow 의 해시레이트 파워를 통한 가속화 방식을 통해 뛰어난 속도를 제공합니다. 또한 MX 노드 시스템은 보안 및 속도를 높여주는 역할로, 이는 21BP 를 선택하는 것과 같이 기존의 시스템에서 속도를 높이기 위해 시도에 대한 솔루션으로, 이전 단계에서 응용한 해시로 형성된 블록체인을 아래와 같이 노드가 형성되어 사용됩니다. Dapp 사용자는 이러한 기능을 사용하여 다양한 지원과 호환이 가능한 소프트웨어를 저비용, 고효율로 개발할 수 있으며, Factor 의 Bootstrap 기술은 기존의 분산장부를 미리 다운 받을 수 있도록 설계되어 부트스트랩의 SHA256 Checksum 을 확인 후 다운받는 형식으로 높은 보안방식으로 전체동기화 과정이 필요 없이 속도를 더욱 향상시킬 수 있는 장점이 있습니다. 이것을 응용하여 향후 미래에 다가올 차세대 네트워크의 수용성을 가능하게 할 뿐만 아니라 블록체인이 가져올 수 있는 미래의 혁신적인 역할을 수행할 수 있도록 하며 기존 블록체인의 방식을 통합적으로 연결시켜 하나의 블록체인으로 만드는 것에 의의가 있습니다.

1)참조 된 블록을 포함하지 않는 포크에서 트랜잭션 재생을 방지합니다.
2)**특허 받은 해시 알고리즘 26 개 이상을 내포**하고 있어 **속도, 보안성, 확장성**이 높습니다.
3)특정 사용자와 지분이 특정 포크에 있다는 것을 네트워크에 알립니다.
4)특정 사용자의 POS 합의 알고리즘 사용을 가능하게 합니다.
5)특정 사용자의 POW 합의 알고리즘 사용을 가능하게 합니다
6)MX Seed node 기술은 특정 지정노드를 먼저 가져와 연결하는 기능을 가지고 있습니다.
7)MX Masternode 기술은 네트워크의 연결유지성, 보안, 속도를 유지 및 증진 시켜주는 기능을 가지고 있습니다.
8)합의알고리즘 POS, POW 와 호환되는 방식으로 운영되며 오라클구간과 연결됩니다. 오라클구간과 연동 시 POS, POW 와 다양한 합의알고리즘을 호환하는 방식으로 작동됩니다.
9)Dapp 사용자는 기존 사용자와 새로운 사용자로 나뉘어지며 상기와 같은 기능들을 사용할 수 있고 기존의 다른 플랫폼의 사용자는 Factor 의 네트워크를 이용하여 Dapp 을 연결시킬 수 있습니다.

1 - 2 분산원장 응용

분산 원장 프레임워크를 이용하여 오라클구간에 존재하고 있는 다양한 프로그램들을 응용하여 분산원장에 접목 시킬 수 있습니다. 따라서 현재 데이터베이스에 응용하고 있는 프로그램 몽고 DB, MySQL, 데이터베이스 관리 시스템 등등에 접목 시키기가 가장 용의하며 분산원장에는 다양한 정보를 담을 수 있기 때문에 기존 스마트 컨트랙트를 응용하는 방법뿐만 아니라 블록체인 원장장부 안에 더 다양한 정보를 담을 수 있습니다. MX 블록체인 분산원장은 아래와 같은 과정을 거쳐서 작동되며 정보를 담아 스마트 컨트랙트가 이루어집니다.

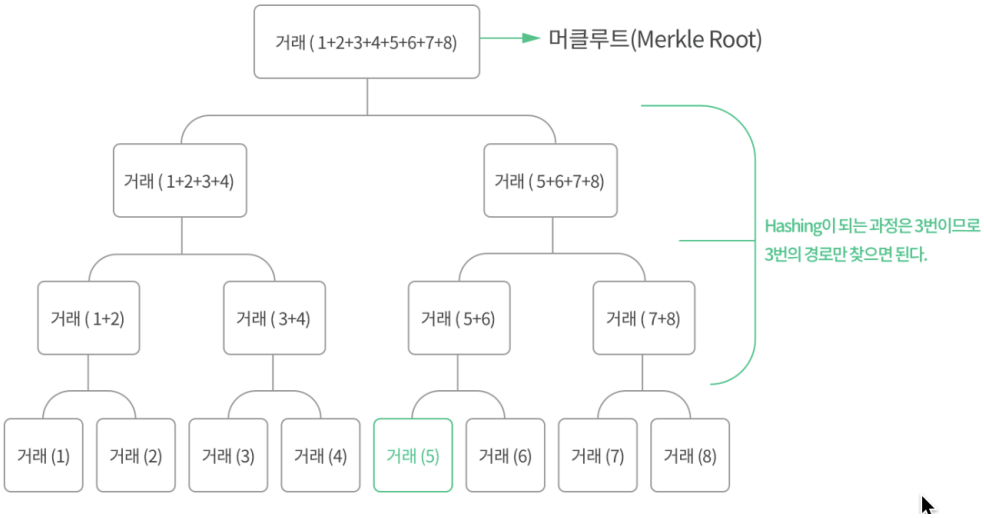


1 - 3 컨트랙트 모델

Factor 블록체인의 컨트랙트 모델은 기존의 이더리움이 가지고 있는 컨트랙트 모델에서 오라클구간인 node.js, java, go 이 세가지를 응용하여 컨트랙트가 이루어지는 방식을 사용했습니다. 하지만 factor 블록체인의 스마트 컨트랙트 모델은 이보다 다양한 지원을 합니다. Node.js, JAVA, go, C++, Python 등과 같은 프로그램을 지원하고 향후 더 다양한 프로그램들을 지원할 예정입니다. 가장 좋은 예시를 들자면 금융사(Visa, Paypal, Mastercard, HSBC, CITI, America Express)에서 이것들을 응용하여 연결할 수 있는 오라클구간을 만들고 이를 기존의 블록체인에 연결하여 사용하는 방식입니다. 하지만 스마트 계약은 오프 체인 데이터 및 API 와 같은 주요 외부 리소스와 연결할 수 없습니다. 따라서 기존의 방식은 중간에 미들웨어나 다양한 서버를 중간에 걸쳐 놓은 형식으로 작동할 수 있도록 만들어져 있습니다. 하지만 이것이 속도저하, 서버오류, 호환성에 대한 다양한 문제를 일으켜 기존의 블록체인을 사용하는 것이 효율성 면에서 굉장히 많이 떨어진다는 문제점이 있습니다. 하지만 Factor 블록체인은 중간의 미들웨어에만 의존 하는 것이 아니라 Factor 블록체인 자체의 성능으로 미들웨이 없이 기존의 프로그램들과 호환 되도록 설계되었습니다.

1 - 4 머클트리 스토리지 모델

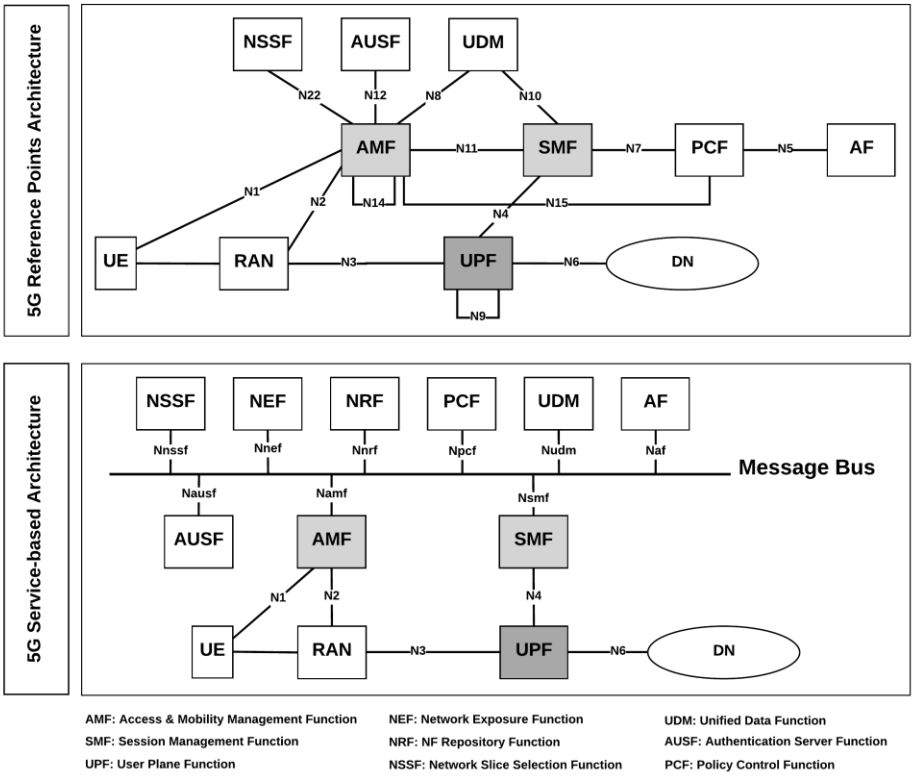
팩터의 머클트리 스토리지 모델은 기존의 머클트리에서 이진수의 값들을 주고받아 이진트리라는 이름으로도 불립니다. 우선 아래와 같은 그림을 참조 합니다.



그림에서의 거래(1+2+3+4+5+6+7+8)값에서 거래 5 번 값을 찾으려면 위의 그림과 같은 과정을 통해 값을 찾아내어 데이터의 위변조 진위 값을 찾아 낼 수 있고 거래량이 기하 급수적으로 늘어나도 특정거래를 찾는 경로는 단순하다는 이점이 있습니다. 기존의 SHA-256 방식은 하나의 해시값을 통하여 하나의 해시값에서 나온 상호작용들을 32 비트로 묶어서 형성해야 하지만 팩터의 블록체인에서는 머클트리에서 각각의 다양한 해시값들이 연결과 정렬 과정을 거쳐 형성됩니다. 이것은 곧 기존의 방식에서는 SHA-256 이 가지고 있는 하나의 해시기능(Funtion)을 지원했다면, Factor 블록체인은 보다 다양한 해시기능을 제공하고 있습니다. 또한 머클트리에서의 검증 과정에서도 특정 거래 경로만 찾는 것이 아니라 각각의 블록이 가진 고유번호(Nonce)값, 시간 값(TimeStamp)들을 찾을 수 있습니다. 이러한 기능들을 추가 사용하여 기존의 머클트리를 검사하는 방식보다 더욱 우수한 기능들을 제공할 수 있습니다.

1 - 5 노드 시스템

Factor 의 노드시스템은 POS Masternode 기능이 첨가된 node 기술을 사용합니다. 이것은 기존의, 채굴만 가능하여 채굴로만 형성되는 해시 파워 레이트에 의존하는 방식을 탈피한 것입니다. 즉 POW, POS 각각의 장점을 둘 다 사용할 수 있는 프로토콜로 작동됩니다. Factor 의 Masternode consensus 는 네트워크의 연결 유지성, 보안성, 속도를 높일 수 있는 역할을 수행합니다. 이것은 곧 Factor 의 특허 MX 노드의 스프레드 방식, Seed Node Technology 기능들을 합친 것으로, 기존의 Masternode 가 단순히 네트워크의 유지성에만 사용되었다면, Factor 의 Node 는 Masternode 와 그 기능들을 생성 유지하지 않아도 기본적인 Node 에 의해 네트워크 유지성에 기여할 수 있는 consensus 입니다. 아래는 5G 네트워크가 가지는 코어 참조 구조 사진입니다.



위의 그림은 다음 세대의 5Generation Network, 즉 5 세대 네트워크의 구조입니다 (5G 네트워크가 어떻게 작동되는지는 생략). 이것이 적용된 스마트폰에서 Factor 프로토콜이 적용된 어플리케이션을 사용할 경우 앞서 언급한 Node 가 Masternode 를 생성하여 기여하지 않아도 더 높은 속도를 가진 노드에서 계속해서 가속화되는, 즉 속도가 계속해서 점진적으로 증가하는 결과를 초래할 것입니다. 이것은 곧, 기존의 노드가 늘어나면 늘어날수록 느려지는 현상을 극복하고, 현재와 미래에 다가올 네트워크환경에서 잘 맞물려 나갈 수 있는 초석이 될 것입니다. 이러한 기술로 인해 네트워크 구축 환경을 위해 사용된 비용을 절감 시킬 수 있는 것입니다.

1 - 6 코어 프로토콜

Factor 의 코어프로토콜은 C 언어로 작성되었으며 윈도우, 리눅스, 맥, 라즈베리파이(지원예정), Android, IOS 등등을 지원합니다. 이것은 Factor 의 코어 프로토콜에서 가지고 있는 다중빌드 시스템을 지원함과 동시에 각각의 운영체제를 지원할 수 있는 다중언어를 지원하기 때문에 가능한 것입니다. Factor 의 코어 프토로콜은 다소 복잡한 과정을 통해 형성됩니다.

1 - 7 사용자 권한관리 프로토콜

MX 블록체인은 공개형 분산장부, 폐쇄형 분산장부 두 가지로 나뉩니다. 폐쇄형 분산장부는 블록체인이 기업에서 사용하는 형식을 이용하여 블록체인 응용기술을 생성할 때 보안성에 대한 부분을 해결할 수 있도록 하기 위함입니다. 폐쇄형 분산장부는 사용자 권한관리 설정에 따라 공개형 분산장부에 접근할 수 있도록 합니다. 이것을 응용하여 다양한 기업, 국가, 그룹, 개인이 용도에 맞게 사용할 수 있도록 하는 데 목적이 있습니다.

1 - 8 분산데이터 교환 프로토콜

Factor 의 교환 프로토콜은 데이터 요청자, 제공자, 에이전트, 소유자를 나누어 보았을 때 각각의 역할에 대해서 분산 실행할 수 있음을 알 수 있습니다. 즉, Factor 블록체인은 이러한 다양한 데이터의 권한을 가지고 있는 사람들이 빠르고, 안전하면서, 사용하기 용이하도록 하는 데 의의를 두고 설계를 했습니다.

1 - 9 어플리케이션 프레임 워크

Dapp(Decentralized application)을 사용하는 사용자들을 위해 Factor 는 Dapp Toolkit 과 One click Dapp 이 존재합니다. 이것은 일반 PC 에서 개발을 할 때 Solidity 언어의 컴파일과 버전이 일치하지 않아 오류가 생기는 현상을 방지하고, 이로 인해 컴파일과 언어를 계속 삭제, 수정, 변경하는 번거로움을 줄일 수 있는 이점이 있습니다. 또한, Ether Solidity 언어만 지원하는 것이 아니라 Qtum, eos, neo 등등과 같이 하나의 툴킷으로 다양한 플랫폼에서 존재하고 있는 언어를 지원할 수 있습니다. One click Dap 을 사용하여 본인이 사용하고자 하는 기능들과 프로그래밍 언어를 몰라도 쉽게 추가 수정 변경할 수 있습니다. 따라서 이렇게 형성된 Dapp 들은 Factor 블록체인 안에서 연결됨과 동시에 다른 플랫폼에서 연결 시도 시, 각각 다른 블록체인 플랫폼에서 같은 언어를 지원하고 호환되는 네트워크를 지원하기에 기존의 다른 플랫폼이 가지고 있었던 비용, 속도, 호환, 확장 문제들을 해결할 수 있고, 결과적으로 모든 블록체인이 연결될 수 있는 기초이자 새로운 블록체인 통합형 어플리케이션 프레임 워크라는 것을 이해할 수 있습니다.

1 - 10 내역관리 모듈

블록체인 분산원장 장부를 응용하여 많은 기술들을 호환 및 연결 할 수 있습니다. 하지만 블록체인 분산원장 장부 안에 담겨있는 정보, 기술, 기록, 인증시스템, 의학시스템 등등 이것을 관리, 제어해줄 모듈이 필요한 것입니다. 예시로 블록체인 원산장부에서 의학정보를 담아서 이것을 MRI 와 연동을 하고 싶다고 가정합니다. MRI 이라는 기계가 사람의 인체의 정보를 찾아내어 이것을 정보로 만들어서 컴퓨터에 전송되는 방식을 사용하지만 그 정보를 통해 사용자는 그 정보에 무엇이 담겨있는지 판단하는 것에 끝이지만, 인체정보를 블록체인 분산원장 장부에 담으면, 그것이 무엇인지 판단하는 것에 그치지 않고 다른 의학정보와 비교 및 실시간으로 의학시술 장비에 MRI 에서 나온 정보를 연동시켜 MRI 결과가 나오자마자 의학시술 장비가 바로 시술이 가능하도록 할 수 있게 합니다. 또한 혈액에 존재하고 있는 DNA, 신원인증, 다양한 질병 분석 또한 많은 비용을 들이지 않고도 검사가 가능할 것입니다. Factor 블록체인은 위와 같은 실생활의 사용성에 입각하여 응용됩니다.

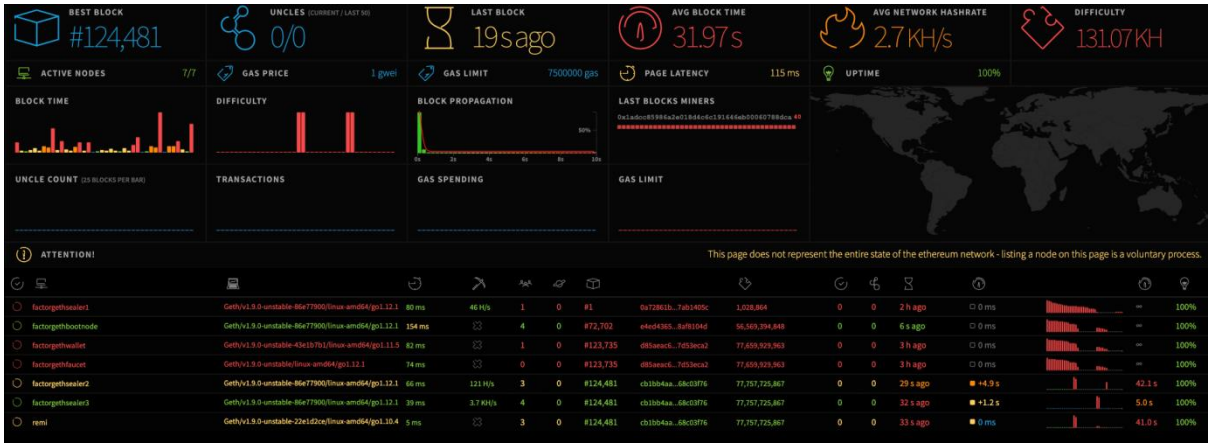
1 - 11 데이터 베이스

이전의 데이터베이스의 의미는 한 사용자가 정보를 수집해 그것을 모아둔 것을 데이터베이스라고 불리는 것이 기본적인 통용 의미였습니다. 이것을 응용하여 현재의 구글, 네이버, 야후와 같은 다양한 검색엔진이 생기고 이것을 통해 우리 삶의 질을 높여 줄 수 있었습니다. 하지만 검색엔진에 존재하고 있는 다양한 정보들이 정보의 양, 질, 정확성이 어떠한 가는 사용자의 판단에 따라 달라질 수 있고 애초에 정보가 잘못될 수 있는 다양한 문제점이 존재합니다. 따라서 블록체인을 응용한 합의 알고리즘, 알고리즘을 통한 정확한 정보선별, 개인이 아닌 집단성 지식 등등을 모두 통틀어 이러한 문제점을 해결하고 나아가 이것을 인공지능 AI 가 이용할 수 있는 정확한 정보체계가 될 수 있으며 이전에 인류가 지식을 책에 담아 이것을 교육하고 실생활에 응용하기까지 소요되었던 시간과 비용을 줄일 수 있는 유일한 해결책으로 작용할 것입니다. Factor 블록체인은 위와 같은 데이터베이스의 개념과 규격을 뛰어넘어 블록체인 사용자의 모든 데이터를 응용함과 동시에 데이터를 생성하고 사용하는 데 있어 기존의 프로그램(몽고 DB, Mysql, MSsql)과 같이 사용할 수 있도록 하는 것을 목표로 하여 설계하였습니다.

2 연결성

2 - 1 Ethereum 연결성

현재 이더리움은 사이드 블록체인, 플라즈마 네트워크, 라이덴 네트워크를 추가하여 확장성 확보를 시도하고, 샤딩을 통하여 속도를 증진시키려는 노력을 하고 있으며, 이것을 통한 다양한 문제를 해결하려고 시도하고 있지만 속도 및 확장성 부분에 있어 부족한 부분이 많이 존재하고 있습니다. 하지만 Factor MX 블록체인의 Dapp 으로 이더리움 기반을 아래의 그림과 같이 연결해 보았을 때 그 결과, Factor 가 가진 기술력은 해시에 내포 되어있는 26 개 이상의 해시와 연동되어 다양한 개발 언어지원을 하며, 기존의 오라클구간의 지원 가능 프로그램들의 범위를 늘릴 수 있을 뿐 아니라, 속도가 증가 한다는 것을 확인 하였습니다. 현존해 있는 이더리움 기반을 사용하는 수많은 코인들 또한 이더리움 과 같이 Factor 블록체인의 DAPP 에 연결 할 수 있습니다. 특히 이더리움은 노드수가 많아질수록 속도가 저하되지만, Factor 블록체인의 특허 MX 노드 스프레드 분사방식 기술로 인해 노드수가 많아질수록 속도가 점진적으로 빨라지는 효과를 볼 수 있습니다.



2 - 2 Eos 연결성

기존의 EOS 는 21 개의 BP 를 선정하고 이에 따른 전체 네트워크 속도를 유지하며 Dapp 을 만들 때 핵심적 요소인 RAM 이라는 컴퓨터 자원성 코인이 존재합니다. Factor 블록체인의 Dapp 으로 EOS 모듈을 연결한다면, 21 개의 BP 가 운영되는 과정에서 소요되는 엄청난 비용을 Factor 블록체인의 Dapp 으로 인해, 이더리움과 같이 연결 하였을 때 속도와 효율성을 증진시키며 21 개의 BP 가 운영되는 고비용, 저효율 방식에서 저비용, 고효율 방식으로 변환 시킬 수 있었습니다. 또한 해시의 연결성 측면에서는 스마트 컨트랙트의 기능들이 오라클구간에 연결되어 사용 될 때 다양한 Solidity 언어를 지원하기에 Factor 의 모듈 같은 Solidity 언어의 버전을 사용하는 블록체인들과 상호호환 되는 구조로써 이러한 방식으로 현존해 있는 Eos 기반을 사용하는 수많은 코인들 또한 Eos 와 같이 Factor 블록체인의 DAPP 에 연결하여 Factor 블록체인과 같은 효력을 볼 수 있을 것 입니다.

2 - 3 Neo 연결성

네오에서 Trinity ICO 를 하는 도중 네오 블록체인이 1816381~1816382 까지 가는데 총 25 분이라는 매우 긴 시간이 걸렸습니다. 이것은 DBFT 의 합의 알고리즘에서 (n-1)/3 개 보다 초과하는 노드가 위변조 문제를 일으키거나 오류를 내면 알고리즘이 작동하지 않는 구조에서, (n-1)/3 을 초과하는 노드가 문제를 일으켜 트랜잭션의 전송을 담당하는 노드에 과부하가 걸려 위와 같은 문제점이 발생하였습니다. Factor MX 블록체인에 Dapp 으로 네오 기반을 연결시킨다면, 위와

같은 합의알고리즘의 결함적인 부분에 대해서 보완 할 수 있을 뿐만 아니라, 특허 MX 노드에 의한 각 노드 간의 전송속도 및 연결성을 높혀줄 수 있는 특징이 있으며, DBFT 합의알고리즘의 문제를 해결함과 동시에 Neo 의 solidity smart contract 의 기능을 통하여 오라클구간에 존재하고 있는 많은 프로그램들을 지원할 수 있으므로 MX 블록체인과 단순히 연결되는 것을 뛰어넘어 네오의 결함을 극복 할 수 있는 역할을 수행할 수 있을 것입니다.

2 - 4 연결성의 의의

MX 블록체인은 위와 같이 기존의 모든 코인들을 Factor Dapp 으로 연결할 수 있으므로, 각각 따로 나뉘어져 있던 블록체인의 응용 가능 기술성을 상호연결 하여 많은 기능들을 사용가능 하게 함으로써 IOT, 무인자동차, 인공지능 등 4 차 산업에서 사용 할 수 있을 뿐만 아니라 속도를 높이려는 라이트닝 블록체인, 플라즈마 네트워크 시도에 대한 비용을 줄이고 높은 확장성을 통하여 산업화에 큰 도움이 될 것입니다. 이와 같은 결과를 통하여 블록 통합연결에 성공하였다고 Factor 기술팀은 주장합니다.

3. 속 도

3 - 1 속도성 해시

GPU, CPU 해시 Funtion 기능이 들어갑니다

팩터의 속도성 해시는 A,B,C 로 나뉘어집니다. 이것은 애플, 윈도우, 리눅스에서 사용될 32bit, 64bit 서로 다른 운영체제에 사용 될 수 있는 근간이 되며 컴퓨터의 성능을 GPU, CPU 다중 스레딩 처리 기능을 지원합니다.

해시 A

1024 비트 상태를 가지고 있으며 512 비트 입력 블록에서 작동합니다. 입력 블록 처리는 다음 세 단계로 구성됩니다.

- 입력 블록 상태의 절반으로 XOR 합니다.

- 해당 상태에 42 라운드 키가 없는 순열 (암호화 기능)을 적용합니다. 이것은 다음의 42 반복으로 구성됩니다.

- 입력을 256 개의 4 비트 블록으로 분할하고 두 개의 4 비트 S- 박스 중 하나를 통해 각각 매핑(mapping) 합니다. 선택은 256 비트 라운드 종속 키 스케줄로 이루어지며, 각 입력 블록을 키 비트와 결합하고 그 결과를 5 → 4 비트 S- 박스를 통해 매핑(mapping) 합니다.
- GF (2 4)에 대해 최대 거리 분리 코드를 사용하여 인접한 4 비트 블록을 혼합 합니다.
- 다음 라운드에서 서로 다른 블록에 인접하도록 4 비트 블록을 설정합니다.
- 입력 블록을 상태 XOR 합니다.

결과 다이제스트는 1024 비트 최종 값에서 첫 번째 224, 256, 384 또는 512 비트입니다. SSE2 명령어 세트를 사용하여 비트 분할 구현에 적합하며 바이트 당 16.8 사이클의 속도를 제공됩니다.

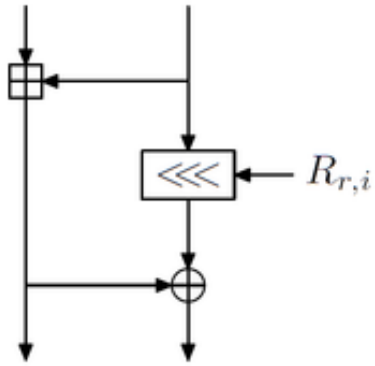
해시 B

B 해시의 내부 상태는 출력 크기의 두 배이며 강력한 메시지 확장을 사용하고 압축 기능에서 수정된 피드 포워드를 사용합니다. 이 작동 모드는 일반적인 공격으로부터 안전 합니다. B 해시의 가장 중요한 구성 요소는 메시지 확장이며, 이는 최소의 거리를 제공하도록 설계되었습니다. 이를 통해 B 해시가 차등 암호 해독을 방지합니다. B 해시는 압축 기능 내부에 작은 스케일의 병렬 처리를 특징으로 하며 이것은 벡터 명령을 사용하여 효율적인 구현을 작성하는 데 사용할 수 있습니다. 이러한 사항은 널리 사용되는 많은 아키텍처(x86 의 SSE , PowerPC 의 Altivec , ARM 의 IwMMXt)에서 사용 할 수 있는 특징이 있습니다.

OS 모드 :	32 비트	64 비트
Core2 에서 의 16 번해시 속도		
16 번해시-256	12 cpb	11 cpb
16 번해시-512	13 cpb	12 cpb

해시 C

256, 512 및 1024 비트의 내부 상태 크기와 임의 출력 크기를 지원합니다. Intel Core 2 Duo 에서 64 비트 모드의 출력 크기에 대해 바이트 당 6.1 높은 사이클을 요구 할 때 해시 C 의 Threefish 기능 핵심은 MIX 함수를 기반으로 합니다. MIX 함수는 상수와 XOR 을 한 번 더 추가하여 두 개의 64 비트 단어를 변환합니다. UBI 체이닝 모드는 입력 체인 값과 임의 길이 입력 문자열을 결합하고 고정 크기 출력을 생성합니다. 해시 C 의 Threefish 기능은 비선형 덧셈 연산과 배타적 논리합의 결합을 사용합니다. 이 기능은 64 비트 프로세서에 최적화되어 있으며 C 해시 용도는 무작위 해시, 병렬 트리 해싱 , 스트림 암호 , 개인 설정 및 키 유도 기능과 같은 선택적 기능을 정의합니다



3 - 2 Seed node

시드노드, 즉 씨앗노드를 의미하며 사용자가 Client 모듈을 사용 할 때에 가장 우선적으로 미리 노드를 찾아내게 하여 연결시켜주는 역할을 합니다. 하지만 먼저 연결된 시드노드가 속도가 느리고 다른 노드가 속도가 더 빠르다고 계산 된다면 시드노드에 먼저 연결 되는 것이 아니라 노드를 유지하고 있는 네트워크에 접속되어 동기화 됩니다. MX Seednode 기술은 이러한 원리를 응용하여 전체 노드의 속도를 계산하고 순차적으로 속도 우선위주로 노드를 연결유지 할 수 있도록 합니다. 이것은 곧 Factor 의 노드가 추가 될수록 비례해서 속도가 빠른 노드가 나타난다면 지속적으로 속도가 증가하는 효과를 가짐과 동시에 속도가 보통이거나 느린 노드가 추가된다면 속도가 느린 노드에 연결이 보류되고 속도가 높은 노드에 먼저 연결되어 빠르게 동기화가 된다는 것이 특징입니다. 속도가 느린 노드가 추가되어도 속도 면에서 우수성을 가질 수 있고, 속도에 관계없이 상호 연결된 노드가 많아질수록 연결성은 높아지는 특성을 가지고 있습니다.

3 - 3 Masternode

기존의 Masternode 는 네트워크 유지성에 기여하면 코인을 보상받는 시스템으로 많이 작용했습니다. 하지만 MX Masternode 는 이보다 더 다양한 기능을 할 수 있습니다.

① 트랜잭션 속도 향상

② 네트워크 공격 방어

③ Seednode 가 스프레드방식으로 연결하는 것을 도와줌

④ Private Send 를 가능하게 함

⑤ 거버넌스(Governance) 기능

⑥ 저전력 고효율 네트워크 기여 기능

3 - 4 MX node

MX node 의 Consensus 에 입각하여 POS, Masternode POS 를 생성하고 운영하는데 아래와 같은 규칙성을 가집니다.

① GPU - CPU 가 포함되어 있지만 GPU 다중스레딩 가속방식을 지원하여 POW 가 지원 할 수 있는 해시레이트 가속방식은 Factor 의 스프레드분사 방식의 근원이 되는 가속방식을 지원합니다.

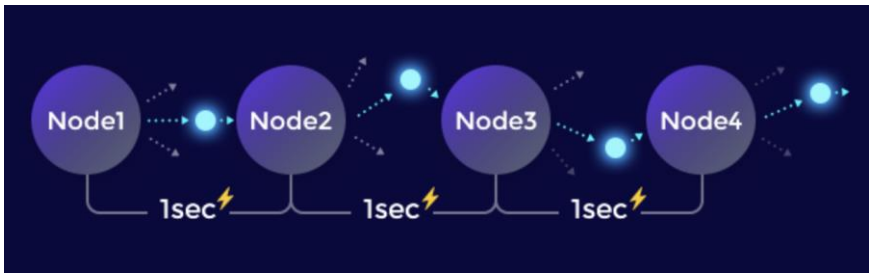
② Seednode - 아래의 [그림 1]과 같이 분사방식으로 분사되는 그림[2]에서 노드가 연결 시에 연결 리스트를 가져와 동기화 시 자동적으로 연결되는 시스템으로 사용되며 Masternode, Normalnode, POW 와 속도를 비교하여 속도우선 순위위주로 연결됩니다. 따라서 연결성에 있어서 속도가 우수하다고 할 수 있습니다.

③ Masternode - 아래의 [그림 1]과 같이 분사방식으로 분사 된 그림[2]에서 노드가 연결시에 MX masternode 는 연결 유지성을 가지므로 동기화 시 연결 할 때의 속도에 대한 기여를 할 수 있습니다.

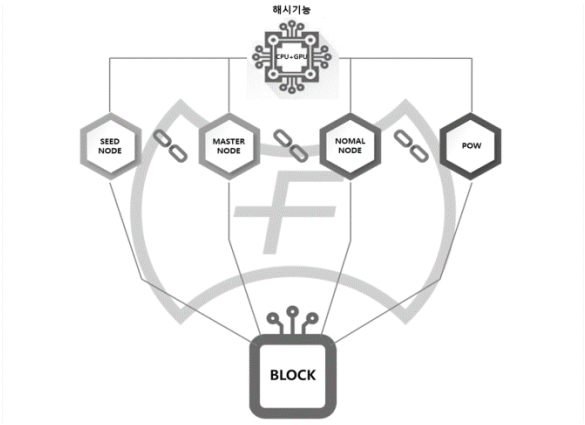
④ POW - 아래의 [그림 1]과 같이 분사방식으로 분사 된 그림[2]에서 Pow 를 실행하고 있는 노드는 연결시에 가속화 방식에 대한 기여를 합니다. 이것은 해시레이트의 파워에 따라 트랜잭션 속도를 높여주는 역할을 함으로써 높은 속도의 노드가 추가될수록 점진적으로 속도가 높아지는 기능을 가지고 있습니다.

⑤ Normalnode - 아래의 [그림 1]과 같이 분사방식으로 분사되는 그림[2]에서 일반적인 사용자가 노드를 형성 할 때 높은 속도성을 가지고 있으면 네트워크의 동기화 시 블록정보를 가져 올 때 속도가 높아지는 것에 대해 기여를 할 수 있습니다.

그림[1] 스프레드 분사방식



그림[2] MX Node system



상세설명 [스프레드 분사방식]
그림[1]과 같은 분사방식에 그림[2] 와 같이 Seednode, Masternode, Normalnode, Pow 4 가지 노드가 존재합니다. 그림[2]의 4 가지 노드가 그림[1]과 같이 동시에 분사가 되며 가장 빠른 노드가 선택됩니다. 우선, 노드가 가진 MX 블록체인의 저전력 GPU 가속방식을 사용하여 Factor 의 Node 시스템의 속도성에서 동기화 가속화를 촉진시켜 주는 역할을 합니다. Seednode 의 기능은 동기화 시 특정 노드리스트를 가져와 자동적인 연결을 해주는 역할을 수행하며 시드노드가 연결됨과 동시에 모든 노드의 속도 계산방식을 통하여 속도 우선순위 위주로 선택 연결되게 하는 역할을 수행합니다. 이러한 방식으로 연결된 후 Masternode 는 노드간의 연결성 유지기능을 통하여 속도 우선순위 위주로 연결 될 때에 모든 노드간의 연결성을 확고히 하며, 최종적으로 POW 가 속도를 가속하는 방식이 되어 모든 노드에서도 속도가 높은 노드와 연결되는 방식을 가지고 있습니다. 따라서 Factor 의 MX 노드는 위와 같은 과정을 노드들이 추가 될 때 마다 속도가 높은 추가 노드수에 정비례하여 속도가 증가하는 특징을 가지고 있습니다. 이는 노드수가 증가할 때 기존의 Eos 21BP 를 Seednode 형식으로 지정해 많은 금액을 투자하여 고비용, 고속도 기능을 낼 수 있는 방식보다 저비용 고효율 방식의 MXnode 가 운영됩니다.

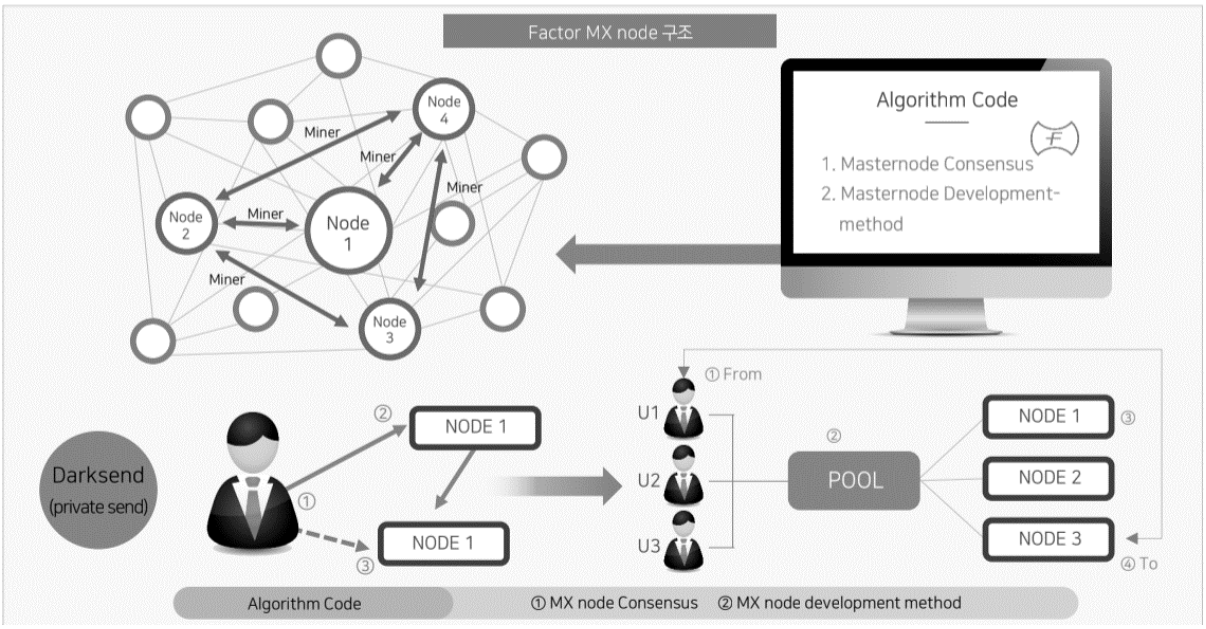
3 - 5 Bootstrap
기존에 존재하고 있는 분산장부를 미리 다운로드 받아 적용시킬 수 있습니다. Bootstrap 은 이전에 존재하는 블록체인 정보를 압축시켜 블록체인정보를 가져오는 과정을 생략 할 수 있으므로 싱크로나이징(Synronizing) 시간을 줄일 수 있는 장점을 가지고 있습니다. MX Bootstrap 기술은, MX 블록체인이 제공하는 SHA-256 함수를 이용하여 파일의 위변조 진위여부를 체크하고 적용 시킬 수 있는 장점이 있습니다.

4. 보안성 또는 안전성

4 - 1 Secp256R1 key
현재 기존에 사용하고 있는 블록체인의 Secp256k1 키의 방식은 ECC(Elliptic Curve Cryptography, 타원곡선 암호기술) 방식에 속하는 ECDSA(Elliptic Curve Digital Signature Algorithm) 암호 알고리즘을 사용하고 있습니다. 기존의 시스템에서는 ECDSA 의 parameter 로 secp256k1 curve 를 사용하며, secp256k1 curve 는 표준으로 제정된, elliptic curve 를 만들기 위한 상수 집합 중의 하나로써 사용됩니다. 즉, ECDSA 는 암호화 방식을 말하는 것이고 secp256k1 키는 이 암호화시킬 수에 대입될 상수집합을 의미하는 것입니다. 따라서 이러한 방식의 RSA 암호기술은 실생활에서 WIFI 비밀번호 형성, ECC in JavaScript, 윈도우 운영체제 Cd-key, SSH-key 등등 다양한 실생활에서 사용되고 있습니다. 하지만 여기서 문제점을 가지는 부분은, 기존의 단어나 용어를 암호화 시켜 비밀키(복호화 시킬수있는 키)와 공개키로 나누어지는 키의 비밀키값을, 양자컴퓨터가 출현하게 된다면, 그 비밀키값을 예측하여 복호화 시킬 수 있는 양자컴퓨터의 연산능력 때문에 ECC, RSA 암호방식이 미래에는 사용할 수 없다는 특징을 가지고 있습니다. 따라서 MX 블록체인은 보안이 더욱 높은 방식의 secp256r1 방식을 사용하여 양자컴퓨터의 연산능력에 대한 보호 해시를 사용합니다.

4 - 2 Private Send
프라이빗 샌드 기능의 원리는 단순합니다. 트랜잭션을 여러 명 거쳐서 보내 후 보낸 사람이 누군지 불명확하게 하여 익명성을 보장받을 수 있는 기능을 합니다. 사용자가 거래를 진행하기 위해 송금할 금액을 지갑에 입력하고 프라이빗 샌드 버튼을 누르면, 당신의 지갑은 입력한 금액을 작은 단위 금액으로 나눕니다. (단위: 0.01, 0.1, 1, 10). 그 다음, 사용자의 지갑은 네트워크상의 특별한 환경 설정을 갖춘 노드들(마스터노드)에게 메시지를 포함해서 전송됩니다. 이때

마스터노드에게 전달된 메세지 값을 입력받고 마스터노드들이 서로 트랜잭션을 랜덤형식으로 주고 받으며 믹싱(Mixing)을 합니다. 각각의 마스터노드는 사용자가 입력한 정보를 섞은 후, 마스터노드 사용자의 지갑에 내용을 전송하고 나누어진 송금하고자 하는 금액을 스스로에게 보내는데, 그 주소는 무작위의 랜덤 형식으로 형성된 주소로 보냅니다. 당신의 자금을 완전히 추적 불가능하게 하기 위해서 당신의 지갑은 이러한 과정을 금액이 나누어 질 수 있는 임의의 수만큼 반복하고, 이러한 과정이 한 번 끝나는 것을 '라운드'라고 부른다. 따라서 가능한 많은 라운드 수를 거쳐 거래가 형성됩니다. 또한 믹싱과정은 사용자가 보이지 않는 곳에서 이루어지며 어떤 경우에도 수신된 마스터노드들은 개인 정보를 수신 받지 않습니다. 따라서 믹싱하는 마스터노드들이 누구인지 추적하기가 불가능합니다.



4 - 3 Merkle Tree 응용보안

앞서 설명한 Factor 의 머클트리 구조에서 이전의 블록 + 현재의 머클트리 + Time Stamp, Nonce 값이 더해져서 Blockheader 를 구성한 하나의 블록이 형성됩니다. 여기서 MX Block 의 Block Time Stamp, Nonce 값을 응용하여 머클트리 구조의 위변조 검사방식에서 한층 더 뛰어난 보안방식을 적용합니다. 타임스탬프는 현재의 시간 값을 UNIX 값으로 변환하여 기록하는 기능을 가지고 있습니다. 여기서 타임 스탬프는 단순히 블록의 시간만 기록하는 것이 아니라 Factor 의 고유블록이라는 문구가 들어갑니다. 따라서 위변조된 블록의 고유문구를 확인하여 검사할 수 있는 기능이 있으며, 지정된 Nonce 값의 변환을 통해 머클트리의 구조에서 해당 데이터를 검사 할 때에 걸리는 과정의 시간을 단축시킬 수 있는 이점이 있습니다. 따라서 MX 블록체인은 기존의 머클트리 구조에만 의존하는 것이 아니라 블록의 위변조 검사의 시간, 보안을 높였습니다.

4 - 4 마스터노드 네트워크 공격보안

DDOS 공격, 네트워크 트래픽 공격 등등의 네트워크 공격문제에 대해서 마스터노드에 의한 네트워크 유지를 통해 높은 보안수준을 제공합니다. 마스터노드는 각각 연결되어있는 노드들에 대한 트래픽분산 시스템, 트래픽 거부 시스템, 거버넌스 시스템을 통하여 특정 공격자가 많은 트레픽을 하나의 노드에 공격할 시에는 다른 노드는 정상적으로 노드를 실행할 수 있으며 전체를 공격한다고 가정하여도 마스터노드의 방화벽, 특정한 포트연결, DNS 서버 트래픽 공격방지, 높은 트래픽 수용력, 거버넌스 시스템으로 네트워크 시스템자체의 수정을 통한 빠른 대응이 가능합니다.

4 - 5 오라클 구간 보안성

블록체인 외부의 환경을 오라클 구간 이라고 명칭 합니다. 이것은 오라클 구간에 존재하고 있는 프로그램들이 블록체인에 연결되며 블록체인에서 실행, 명령되는 것을 실질적인 외부환경에서 작동되게 하는 것입니다. 오라클 구간에 존재하고 있는 JAVA, NodeJS, Mongodb, Mysql 등등의 프로그램이 연결될 때 Factor 블록체인 안에서 보안성, 작동성을 검사 후 실행됩니다. 따라서 여러 가지의 종류 프로그램이 MX 블록체인과 연결될 때 각 프로그램 간 필터링 과정을 통하여 연결되는 것입니다. 하지만 향후 MXblock 연동 시 안전성이 확인된 프로그램들은 향후 Dapp Toolkit 에서 공식적으로 지원할 예정입니다.

4 - 6 보안성 해시

팩터블록체인의 보안성해시 D, E 는 아래와 같은 보안성 해시 기능을 합니다.

- 해시 D

SHA-3 에 대한 D 번 해시-f [1600] 인 블록 변환 f 는 **XOR** , **AND** 및 NOT 연산을 사용하는 순열이며 소프트웨어 및 하드웨어에서 쉽게 구현할 수 있도록 설계되었습니다.

이는 2 의 워드 크기, $w = 2^{\ell}$ 비트에 대해 정의됩니다. 주요 SHA-3 제출은 64 비트 단어, $\ell = 6$ 을 사용 합니다.

상태가 고려 될 수 있는 5 × 5 × w 비트들의 배열. 리틀 엔디안 비트 번호 매김 규약과 행 주요 인덱싱을 사용하여 $a[i][j][k]$ 가 입력의 비트 $(5i + j) \times w + k$ 라고 합시다 . 즉, i 행 선택한다 j 열을, 그리고 k 는 비트를 의미합니다.

인덱스 산술은 처음 두 차원은 모듈로 5 로 수행 되고 세 번째 차원은 모듈로 w 로 수행됩니다.

기본 블록 교체 기능은 12 + 2 ℓ 의 5 단계로 구성됩니다.

양자공격에 대한 보안

일반적인 컴퓨터 (Grover 알고리즘)는 퀀텀 컴퓨터가 $\sqrt{2^{d}} = 2^{d/2}$ 의 구조화 된 프리 이미지 공격을 수행 할 수 있는 반면 고전적 brute-force 공격은 2^{d} 가 필요합니다 . 구조화 된 프리 이미지 공격은 두 번째 프리 이미지 공격과 충돌공격을 의미 합니다. 양자 컴퓨터는 또한 수행 특정지정 날짜공격에 따라서 충돌내성을 파괴, $\sqrt[3]{2^{D}} = 2^{D/3}$. 최대 강도는 $c/2$ 이며 SHA-3 의 양자 보안에 대해 다음과 같은 상위 경계를 제공합니다.

예	비트 단위의 보안 강도			
	충돌 (Brassard et al.)	충돌 (번스타인)	프리 이미지	두 번째 이미지
SHA3-224 (M)	74/3/3	112	112	112
SHA3-256 (M)	85 1/3	128	128	128
SHA3-384 (M)	128 자	192	192	192

SHA3-512 (*M*)	170/3/3	256	256	256
SHAKE128 (*M* , *d*)	분 (*d* / 3,128)	분 (*d* / 2,128)	≥min (*d* / 2,128)	분 (*d* / 2,128)
SHAKE256 (*M* , *d*)	분 (*d* / 3,256)	분 (*d* / 2,256)	≥min (*d* / 2,256)	분 (*d* / 2,256)

- 해시 E

E 해시-128 뿐만 아니라 원래의 E 해시는 128 비트 결과가 너무 작고 설계 상 약점 때문에 (원래 E 번 해시의 경우) 안전하기 때문에 안전하지 않은 것으로 간주됩니다. 256 및 320 비트 버전의 E 해시는 각각 E 해시-128 및 E 번 해시-160 과 동일한 수준의 보안을 제공합니다. 보안 수준은 충분하지만 더 긴 해시 결과가 필요한 응용 프로그램 용으로 설계되었습니다.

E 해시-160 (RIPE 메시지 다이제스트 라고도 함)는 일반적으로 40 자리의 16 진수로 표시됩니다. 다음은 43 바이트 ASCII 입력 및 해당 E 해시-160 해시를 보여줍니다.

```
E 해시-160("The quick brown fox jumps over the lazy dog") =
 37f332f68db77bd9d7edd4969571ad671cf9dd3b
```

E 해시-160 은 암호화 해시 함수 (작은 변경, 예를 들어 d 를 c 로 변경 하면 완전히 다른 해시가 됩니다) 의 원하는 애벌랜치 효과와 함께 동작합니다.

```
E 해시-160("The quick brown fox jumps over the lazy cog") =
 132072df690933835eb8b6ad0b77e7b6f14acad7
```

길이가 0 인 문자열의 해시 값은 다음과 같습니다.

```
E 해시-160("") =
 9c1185a5c5e9fc54612808977ee8f548b2258d31
```

5. 확장성

5 - 1 Factor ledger Module (팩터렛저 모듈)

팩터렛저 모듈은 다양한 페브릭 기능들을 제공하고 있습니다. 주로 현재 제공 가능한 기능은 동물추적, 채권, 자동차경매, 디지털재산, 기금개설, 무역거래 신용장, 게임, 음식관리, 신원인증, 거래소, 자동차 라이프 사이클 주기관리, 자동차 정비주기 관리 등등과 같이 실생활에서 다양한 용도를 가지고 잇습니다. 팩터렛저는 실생활의 사용자가 블록체인을 알지 못해도 자신의 사업에서 구축하려는 시스템을 보다 손쉽게 구축할 수 있도록 해주며 블록체인의 기술을 응용해 비용절감, 높은 기술력, 다양한 BA 모델 네트워크를 제공하며, 낮은 비용으로 고효율의 효과를 갖는 장점이 있습니다. 이것을 응용하여 기존의 프로그램에 적용하여 다양한 응용이 가능합니다.

5 - 2 Graphene (그래핀)

Factor 그래핀 모듈은 가격 안정 암호화폐, 분산화된 자산의 교환, 산업에서 요구하는 성능과 확장성, 능동적인 계정의 허용, 계정의 모집과 계획된 지불, 보상프로그램, 유저자산, 이해관계자들이 승인한 프로젝트 펀딩, 변환 가능한 계정, DPOS 합의 일치를 제공 합니다. 유저자산(UIA) 시스템에서는 무형의 자산을 토큰에 접목시켜 사용할 수 있고 종류별로 자산을 나눌 수 있습니다. 또한 기업에서 사용하는 블록체인이 비공개 형식으로 사용되기를 원한다면 능동적인 계정의 허용 기능을 통해 승인된 사용자만 사용할 수 있고 다양한 비즈니스 모델(BA)을 형성하기에 적절한 특징을 가지고 있습니다. 그래핀 모듈은 위와 같은 기능을 이용하여 실생활에서 회사가 할 수 있는 순기능, 운영, 관리의 역할을 총체적으로 가능하게끔 합니다. 따라서 기존의 블록체인이 기업으로 운영되기 어려움 점을 해결하는 것에 의의가 있습니다.

5 - 3 금융

Swift 시스템의 분산장부

IBM Mainframe (통합은행전산 시스템)

HP UNIX (차세대 시스템)

HP Superdome (통합차세대 시스템)

IBM UNIX (WINS)

제 1 금융, 제 2 금융, 제 3 금융 등

금융권에서 사용하는 Swift 시스템을 위와 같이 현재사용 중인 금융권 시스템 규격에 Factor Dapp 과 Toolkit 을 사용하여 금융권에 활용 할 수 있도록 합니다.

5 - 4 IOT

AMQP(Advanced Message Queuing Protocol)

CoAP(Constrained Application Protocol)

DDS(Data Distribution Service)

JSON-LD(JavaScript Object Notation for Linked Data)

MQ Telemetry Transport

Near-field Communication

Supervisory Control and Data Acquisition

6LoWPAN, HomeKit, IoTivity, LoRaWAN, Zigbee

5 - 5 공공기관

민원서류, 국민투표, 신원증명 확인용 민원등기 서류 등에 대하여 Factor 블록체인은 많은 비용절감과 함께 보안성을 높여주는 역할로 사용 될 것입니다.

5 - 6 물류, 유통

CRM, SCM, ERP, Logistic System

제품간의 물류이동에서 QR 코드 조회가 가능하며, 유통시스템에서 분산화된 장부로써 중앙화된 방식으로 확인 하는 것이 아닌 일부만 확인을 해도 조회가 가능합니다.

5 - 7 제조, 생산

블록체인은 제조와 생산에서 사용되는 로봇에 가장 먼저 적용시킬 수 있는 대상이 될 것입니다. 이것은 로봇이 가지고 있는 언어수행 기능들을 블록체인 분산장부에 적용시켜 작동 될 수 있도록 하는 것을 의미합니다.

6.Dapp(생태계)

6 - 1 Factor Dapp

① 요약 및 생태계 개요
② 기술 생태계
③ 거버넌스 생태계
④ 어플리케이션 생태계
⑤ 운영방식 및 참여방법

6 - 2 Factor ToolKit

Factoremix 와 Oneclick Dapp 이 지원되고 있습니다. Factoremix 는 개발에 팩터의 Ethereuem, Neo, Eos 모듈에 사용되는 언어를 통합적으로 지원합니다. 또한 Oneclick Dapp 을 사용하여 Dapp 구축사용자가 버튼만 누르면 Dapp 이 형성되는 방식으로, Dapp 구축시에 뛰어난 편리성과 Dapp 에 응용될 사용 언어를 조금만 숙지하더라도 Dapp 응용형성이 가능하도록 설계가 되어있습니다
앞으로 향후 추가될 오라클 구간에서 블록체인과 호환 될 수 있는 다양한 프로그램이 Factor ToolKit 에 추가될 예정이며 이것을 응용하여, Dapp 개발자가 아니라 일반적인 프로그래머가 블록체인과 연결하여 응용 개발 하더라도 뛰어난 확장성과 편리성을 제공 할 수 있는 특징이 있습니다.

6 - 3 지원언어 및 프로그램 (오라클 구간)

① Go 언어

구글에서 개발했으며 현재 블록체인과 연결할 때 사용되는 요소 중 하나로써 언어의 설계와 Go 의 문법은 대체로 C 와 비슷합니다. 코드 블록들은 중괄호로 둘러싸고 for, switch, if 를 포함한 일반적인 제어구조를 가지고 있습니다. C 와 다르게, 한 라인 끝의 세미콜론은 필수가 아닌 옵션 입니다. 변수 선언은 다르게 작성되는 옵션입니다. 형 변환은 명시적으로 하며 병행성 프로그래밍을 다루기 위해 go 와 select 키워드가 사용됩니다. 새로운 타입은 map, 유니코드 문자열, 배열 slice, 그리고 내부 쓰레드 통신을 위한 채널(channel)이 있으며 Go 는 그리 좋지 않은 하드웨어에서도 빠르게 컴파일 될 수 있도록 디자인되었습니다. 따라서 저사양의 컴퓨터에서 컴파일하여 파일이 위변조, 감염되지 않는 순수한 Client 로 사용이 가능합니다. Go 는 가비지 컬렉션이 되는 언어이며 병행성(concurrency)과 관련된 Go 의 구조적인 규칙들(channel 과 선택적인 channel input 들)은 Tony Hoare 의 CSP 로부터 가져온 것입니다. C++, JAVA 에 있는 기능들 중 타입 상속, 제너릭, 표명(assertion), 메서드 오버로딩, 포인터 연산은 Go 에서 포함하고 있지 않습니다. 따라서 Go 언어의 개발언어로써 부족한 부분을 다른 지원 가능한 오라클 구간의 언어로 보완할 수 있습니다.

② JAVA

객체 지향 프로그래밍(Object-Oriented Programming, OOP)이며 지향 프로그래밍은 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 "객체"들의 단위를 파악하는 것으로 사용됩니다. 각각의 객채는 메시지, 데이터를 주고 받으며 프로그램 변경을 용이하게 해주는 기능을 합니다. 따라서 대규모 소프트웨어 개발에 주로 사용되며 개발 유지, 보수가 간편합니다. 직관적인 코드 분석을 가능하게 하지만 지나친 객채화 경향은 실제 개발에 접목할 때 개발의 필요한 요소들을 많이 적용시키지 못하는 단점이 있습니다. 자바에서는 javac.exe 프로그램이 컴파일 기능의 역할을 하며 자바는 자바 실행 환경(JRE: Java Runtime Environment)이 설치되어 있는 모든 운영체제에서 실행 가능합니다. JAVA version 8 부터 람다식(Lambda Expressions)을 지원하며 컬렉션의 요소를 필터링, 매핑, 집계 처리하는데 쉬워지고, 코드가 매우 간결한 함수적 프로그래밍을 지원합니다 C++은 메모리에 생성된 객체를 제거하기 위해 개발자가 직접 코드를 작성해야 되지만 자동으로 사용하지 않는 객체를 제거하는 기능을 가지고 있습니다. 이것으로 인한 다양한 어플리케이션을 구현할 수 있으며 멀티 스레드를 쉽게 구현할 수 있는 특징이 있습니다.

③ Node.js

기존의 자바스트립트의 문제성에 입각하여 만든 V8 엔진을 사용하고 있습니다. 이것은 구글에 의해 지속적으로 업데이트 및 기능 추가가 될 것이며, 이벤트의 Driven 방식을 사용하고 있습니다. 이것은 사용자가 이벤트를 발생시켰을 때 입력장치로 데이터를 전송했을 때에만 작동하는 방식입니다. 발생한 이벤트에 대해서만 웹서버가 '연결'을 해주기 때문에 자원을 최소화 할 수 있고 대부분의 웹서버는 사용자가 이벤트를 발생하기 전까지 계속 작동하고 있으므로 대기시간, 메모리 소비량이 증가하게 됩니다. 또한 Non-blocking I/O 방식을 사용하며 기존의 방식은 Read/Write 이벤트가 발생하게 되면 해당모듈을 모두 점유하여 유저 프로세스의 모든 작업을 중단하게 되고 해당 이벤트에 모든 메모리를 사용하게 되지만 Non-blocking I/O 는 이벤트가 시작하자마자 모듈을 변환시켜 다른 작업 하도록 준비상태가 될 수 있게끔 합니다. 따라서 속도가 기존의 동기식보다 빠르며 메모리를 덜 자치하게 됩니다. 싱글 스레드를 지원하며 클라이언트와 서버에서의 언어가 동일하다는 장점이 있습니다.

④ C, C++언어

C 라는 언어는 매우 심플하면서도 소프트웨어 구성의 최소단위인 bit 부터 시작해서 메모리 관리, 그리고 고급 개념인 OOP 까지 근접하여 사용할 수 있습니다. 실제로 가장 기저에 놓인 OS API 는 오늘날 플랫폼을 불문하고 거의 다 C 언어로 되어있고, 그 외에도 대부분의 인프라가 되는 소프트웨어들은 C 로 만들어진 후 타 언어로의 바인딩을 제공하는 식입니다. 낮은 단계에서부터 단계를 높여가며 관찰을 해보면, 머신코드는 머신에 따라 달라지고, 어셈블리어도 Intel/AT&T 등 문법에 따라 몇 가지 버전이 존재하지만, 그 위쪽에서 결국 C 언어로 대통합이 이루어집니다. 그리고, C 언어 위쪽으로 가면 다시 C++ / Java / C# / Objective-C / Python 등으로 다양하게 갈라지면서 즉, 두 개의 원뿔을 꼭지점끼리 붙여놓은 double cone 형태이며, 꼭지점 부분에 C 언어가 존재하는 형태이니 이것만으로도 C 언어의 중요성은 충분히 알 수 있습니다. 그렇기 때문에 실제로 C 언어를 쓰지 않고 타 고급언어를 사용하더라도 연동해서 사용할 수 있고 이 경우, 많은 수의 언어가 C 언어와의 FFI 를 제공하는 사례가 있습니다.

7. 게임

7 - 1 인리얼 엔진
7 - 2 유니티 엔진
7 - 3 하복 엔진
7 - 4 크라이 엔진
7 - 5 주피터 엔진
7 - 6 게임브리오 엔진
7 - 7 소스 엔진

8.보안 솔루션

8 - 1 Factor 백신
8 - 2 Factor Online Security
8 - 3 Factor 기업형 솔루션

9. 용어

MX(Multi X)블록체인

MX 분산원장

ECDSA(Elliptic Curve Digital Signature Algorithm)
ECC(Elliptic Curve Cryptography) 방식에 속하는 디지털 서명 용도로 사용되는 암호 알고리즘입니다. ECC 는 RSA 암호방식에 대한 대안으로 1985 년도에 제안된 방식입니다. 암호키의 길이가 길면 보안은 강화 되지만, 암호연산 속도가 감소되기 때문에, 보안을 강화하기 위해, RSA 방식의 암호키 길이를 늘리는 대신에 ECC 방식을 사용하는 추세에 있습니다.

secp256k1
ECDSA 의 parameter 로 secp256k1 curve 를 사용합니다. secp256k1 curve 는 표준으로 제정되어 있으며, elliptic curve 를 만들기 위한 상수 집합 입니다. 이것을 "Elliptic curve 256-bit domain parameter" 라고 부릅니다. sec(Standard for Efficient Cryptography) + p(Parameter p over Fp) + 256(Field Size p 의 bit 수) + k(Koblitz curve 변형) + 1(sequence number) 로 구성되어 있습니다.

Secp256r1
ECDSA 의 parameter 로 secp256r1 random curve 를 사용합니다. secp256r1 처럼, k 대신 r(Random Parameter)를 사용하는 Curve 를 사용합니다.

MX 노드
MX 노드는 네트워크 연결을 유지 하므로, 소수의 채굴 집중화 현상으로 인한 문제점을 해결해주는 역할을 하며, 이것은 채굴서버와 노드의 상호작용을 통해 더욱 더 견고하고 안전한 분산형 블록체인을 형성할 수 있습니다.

Merkletree(머클트리)
Factor MX 블록은 상호작용하는 해시트리, 즉 Merkletree 를 형성하며 데이터 전체가 아닌 일부만 검증하고자 할 때에도 자식노드 가운데 하나의 해시 값을 알면 그 노드의 모든 자식노드에 대해 데이터를 검증 할 수 있는 특징을 가지고 있으며 형성된 블록은 지속적으로 이전 블록들과 연결되며 생성되기 때문에 분산화된 장부, 즉 강력한 보안성을 갖춘 블록체인이 형성됩니다.

Hash(해시)
해시함수는 임의의 길이를 갖는 임의의 데이터에 대해 고정된 길이의 데이터로 매핑하는 함수를 말합니다. 이러한 해시 함수를 적용하여 나온 고정된 길이의 값을 해시값 이라고 합니다. 이 값은 또한 해시 코드, 해시섬(sum), 체크섬 등으로도 불립니다.
해시 함수는 보통 그리 복잡하지 않은 알고리즘으로 구현되기 때문에, 상대적으로 CPU, 메모리 같은 시스템 자원을 덜 소모하는 특성이 있습니다. 그리고 같은 입력값에 대해서는 같은 출력값이 보장되며, 이 출력값은 가능한 한 고른 범위에 균일하게 분포하는 특성이 있습니다. 특수 목적용으로 해시값을 생성하는 원본과 별도의 값을 입력 받아서 같은 입력에 대해 다른 출력값을 가지게 하는 해시 함수도 존재합니다.

해시 함수는 보통 입력값의 범위보다 출력값의 범위가 좁은 경우가 많기 때문에 입력이 다름에도 불구하고 드물게 동일한 값이 출력되는 경우도 존재합니다. 자세한 원리는 비둘기집 원리를 사용합니다. 이러한 경우를 '충돌' 한다고 합니다. 원칙적으로 해시 함수는 이런 어쩔 수 없는 충돌을 제외하고 의도적으로 충돌을 계산해낼 수 없어야 합니다.

이러한 특성에 힘입어 다양한 목적에 맞게 설계된 해시 함수가 존재하며 다음과 같은 다양한 분야에서 매우 유용하게 사용됩니다.

- 자료구조
- 해시 테이블 (또는 해시 맵)
- 해시셋(set)
- 블룸필터 (Bloom filter)
- 캐시
- 중복 레코드 검색
- 유사 레코드 검색
- 유사 부분 문자열 검색
- 기하학적 해시
- 변조 탐지/에러 검출

근래에 나오는 언어들은 기본 라이브러리에 해시 함수가 포함되어 있는 경우가 많아서 굳이 구현할 필요없이 바로 바로 해시 값을 추출해서 사용할 수 있습니다. 다만 좀 오래된 언어들은 확장 라이브러리를 설치하거나 직접 구현하는 식으로 해결해야 합니다. 파이썬의 경우에도 사전(Dictionary)에 클래스를 넣기 위해서는 해시 함수를 구현해야 하는데, 해시와 함께 비교함수(cmp)도 구현해야 합니다. 만약 해시 함수가 구현되어 있지 않다면 그 객체의 주소값을 해시 값으로 대체합니다.

유명한 해시 알고리즘으로 Message-Digest Algorithm(MD)과 Secure Hash Algorithm(SHA) 등이 있습니다. 각 알고리즘은 심각한 해시 충돌 문제 등으로 인해 해시 함수를 개선하며 발표된 순서대로 MDn, SHA-n 식으로 넘버링 됩니다. 다만 SHA-2 는 예외로, SHA-256, SHA-512 를 함께 SHA-2 족(SHA-2 family)이라고 부릅니다. 2014 년 기준으로 최신 버전은 각각 MD6, SHA-3 이나 보통은 자신이 사용하는 언어에서 제공하는 라이브러리에 포함된 기본 해시 함수를 사용하는 편입니다.

ECC(타원곡선 암호기술)

ECC 란 "유한체(Finite Field) 상의 타원곡선(Elliptic Curve)" ---영어로 표현하면 "Elliptic Curve over Finite Field" 의 수학적 성질을 이용한 암호 기술입니다. ECC 는 RSA 암호방식에 대한 대안으로 1985 년도에 제안된 방식입니다. 암호키의 길이(bit 수로 표시)가 길면 보안(암호 해독에 소요되는 시간이 아주 많이 필요함)은 강화 되지만, 암호연산 속도가 느려지게 때문에, 보안을 강화하기 위해, RSA 방식을 사용하며 암호키 길이를 늘리는 대신에, ECC 방식을 사용하는 추세에 있습니다. 즉 ECC 를 사용하면, 적은 bit 수의 암호키로 동일한 암호성능(암호 해독에 소요되는 시간)을 나타내기 때문입니다. 예를 들면, 3072-bit RSA 와 256-bit ECC 의 암호성능이 동일하다고 합니다. bit 수가 적으면 연산이 보다 빠르게 처리될 수 있기 때문에 암호연산 성능이 좋아집니다. 다른 관점에서 이야기하면, 암호연산 장치의 CPU 성능이 낮아도, 암호성능을 유지할 수 있다는 것입니다.

10. 결론

본 백서의 내용에서 지금까지 살펴본 바와 같이 팩터 MX 블록체인의 기술을 요약해서 아래와 같이 정리해 보겠습니다.

아 래

1) 팩터 MX 블록체인은 메인블록의 secp256r1방식으로 26개 이상의 새로운 해시알고리즘 기능을 개발하여 탑재하였으며 이후 새로운 신기술인 양자컴퓨터가 나온다면 추가 업데이트 후, 탑재 가능하도록 설계되어 있습니다.

2) 팩터 MX 블록체인의 기술은 기존의 코인들과 연결될 뿐만 아니라 호환이 가능하도록 설계 되어 있습니다

3) 팩터 MX 블록체인 또한 비트코인과 같은 채굴방식으로 반감기 기능은 있지만, 높은 해시기능으로 전력소비량은 비트코인의 100분의 1 수준으로 감소시켜 채굴됩니다.

4) Pow, Pos 기능 호환이 가능하며 마스터 노드 채굴이 가능하도록 설계되어 누구나 채굴 가능합니다.

5) MX 블록체인의 노드 선택 방식은 특허 받은 스프레드 분사시스템으로 우선순위 노드선택 방식으로 설계되어 기존의 노드선택 방식보다 속도가 월등히 우수할 수 있습니다.

6) 블록체인의 분산장부에 양자역학의 보안기능이 탑재되어 오라클 구간에도 보안이 우수하다 할 수 있습니다.

7) MX 블록체인은 자체 개발한 메인 블록체인으로 팩터댑(DApp)이 있으므로 수많은 확장성을 보일 수 있습니다.

8) MX 블록체인은 팩터댑으로 툴킷을 개발하여 새로운 개발자뿐만 아니라 사용자 누구나 쉽게 접근하여 활용할 수 있도록 설계 되어있습니다.

9) 팩터 MX 블록체인은 7종류의 게임을 보유하고 있으며 팩터댑으로 MX 블록체인과 연결하였습니다.

10) 팩터 MX 블록체인이 MX 블록체인과 인공지능을 결합하여 모든 산업화 시스템에 활용될 수 있도록 개발하여 생산성을 높이고 비용은 절감하며 안전한 산업 사회가 될 수 있도록 팩터 연구팀은 꾸준히 노력할 것입니다.

면책조항

본 백서에 제시된 내용은 정보 제공을 목적으로만 작성된 것이며 이 문서의 진술에 의존해서는 안됩니다.
본 백서에 명시된 정보로부터 발생하는 모든 법적 책임을 지지 않습니다. 특히 본 백서에 명시된 "개발기술"은 변경될 수 있으며, 코인의 성능과 수익 관련 어떠한 진술에도 연관되지 않습니다. 본 백서에 실린 정보를 감독하거나 승인하는 규제기관은 없습니다. 따라서 규제 요구사항 또는 관할 규칙에 따라 어떠한 필요한 법적인 조치도 받아들여지지 않습니다.
본 백서의 발행, 배포, 또는 보급에 있어서 준거법과 규제 요건 또한 규칙에 얽매이지 않습니다.
백서에 있는 정보는 변경될 수 있습니다. 팩터 블록체인은 꾸준히 연구하므로 추후 본 백서의 변경사항들은 "수정 버전" 부분에서 확인할 수 있습니다.