Teknikal na Whitepaper

Pangkat Harmony

Bersyon 2.0

1. Panimula

Mula pa nung paglalathala ng Bitcoin whitepaper noong 2008, ang konsepto ng blockchain ay kumalat sa buong mundo. Habang ang desentralisadong pera at mga aplikasyon ay nagiging mahusay na publicized na mga ideya, ang mga limitasyon sa disenyo ay hinamon ang pangunahing paghahangad ng Bitcoin. Ang orihinal na Bitcoin blockchain ay dinisenyo bilang isang peer-to-peer na sistema ng pagbabayad [13] na nagpapahintulot sa mga tao na maglipat ng halaga nang walang tagapamagitan tulad ng mga bangko o mga processor ng pagbabayad. Gayunpaman, nang nakakuha ng pagkilala ang Bitcoin, ang bottleneck sa pagganap nito ay naging maliwanag dahil sa limitadong paghahatid nito ng ~ 7 transaksyon kada segundo (TPS), at ang gastos nito bilang isang sistema ng pagbabayad ay humahantong sa kamahalan.

Noong 2014, Buterin et. al. [27] nagpanukala ng isang bagong imprastraktura ng blockchain na tinatawag na Ethereum, na nagpapagana ng nilikha ng mga developer sa iba't ibang uri ng mga aplikasyon ng blockchain gamit ang "smart contracts." Gayunpaman, hindi nalutas ng Ethereum ang problema sa scalability at, kasama ang ~ 15 TPS, nabigong masuportahan ang mataas na throughput ng mga application tulad ng paglalaro o desentralisadong palitan.

Dahil sa mga limitasyon sa pagganap ng Ethereum at Bitcoin, maraming mga proyekto sa blockchain ang nagpanukala ng iba't ibang mga solusyon [3,4,5,6,7,8,9,10,24,25] na sumubok na mapataas ang throughput ng transaksyon. Ang iba't ibang blockchain [3,4,5,6,24,25] ay iminungkahi na palitan ang kasunduan ng Proof-of-Work (PoW) consensus sa kasunduan ng Proof-of-Stake (PoS) consensus. Ang iba pang mga blockchain tulad ng EOS ay gumagamit ng Delegated Proof of Stake (DPoS), kung saan ang mga nagmumungkahi ng block ay inihalal sa pamamagitan ng pagboto sa halip na sa pamamagitan ng proseso ng algorithm sa on-chain. Ang mga proyektong tulad ng IOTA ay pinalitan ang istraktura ng datos ng chain-of-blocks na may istraktura ng data ng DAG (Directed Acyclic Graph), na pumipigil sa limitasyon ng sunud-sunod na pagproseso ng mga transaksyon.

Gayunpaman, ang mga iminungkahing solusyon ay hindi maaaring gumawa ng makabuluhang mga natamo sa pagganap nang hindi isinakripisyo ang iba pang mga kritikal na aspeto, tulad ng seguridad at desentralisasyon. Ang solusyon sa scalability na parehong nagpapanatili ng seguridad at desentralisasyon ay ang sharding, na lumilikha ng maramihang mga grupo (i.e. shards) ng mga validator at hinahayaan silang iproseso ang mga transaksyon na kasabay nito. Bilang resulta, ang kabuuang paghahatid ng transaksyon ay tataas nang linearly habang lumalaki ang bilang ng shards. Zilliqa [12] ang unang public blockchain na iminungkahi upang matugunan ang problema sa scalability na may sharding. Gayunpaman, ang paglapit sa pag-sharding ni Zilliqa ay nahati sa dalawang paraan. Una, hindi nito hinati ang pag-iimbakan ng datos ng blockchain (state sharding). Pinipigilan nito ang mga makina na may limitadong

mga mapagkukunan mula sa pakikilahok sa network, kaya binabawasan ang desentralisasyon. Ikalawa, ang proseso ng sharding ni Zilliqa ay madaling kapitan sa isang pag-atake ng isang solong shard dahil sa pag-asa nito sa PoW bilang mekanismo ng randomness sa henerasyon nito.

Ipinakilala namin ang Harmony, ang susunod na henerasyon ng sharding-based na blockchain na ganap na nasusukat, mapagkatiwalaan at matipid sa enerhiya. Ang Harmony ay tumutugon sa mga problemang umiiral sa blockchain sa pamamagitan ng pagsasama-sama ng mga pinakamahusay na resulta ng pananaliksik at pagsasanay sa engineering sa isang mahusay na ayos ng sistema. Sa partikular, ang Harmony ay gumagawa ng mga breakthrough sa mga sumusunod na aspeto:

- **Ganap na nasusukat:** Ang mga shard ng Harmony ay hindi lamang ang komunikasyon sa network at pagpapatunay ng transaksyon tulad ng Zilliqa, kundi pati rin mga shard ng blockchain state. Ginagawa nito ang Harmony na isang ganap na nasusukat na blockchain.
- **Ligtas na Sharding:** Ang mga proseso ng sharding ng Harmony ay siguradong ligtas salamat sa proseso ng distributed randomness generation (DRG) kung saan ay hindi nahuhulaan, walang kinikilingan, napapatunayan at nasusukat. Ang network rin ng Harmony ay nareshards sa isang di-interuptibong paraan upang maiwasan ang dahan-dahan na paggamit ng mga kaaway ng byzantine.
- **Matipid at Mabilis na Kasunduan:** Hindi tulad ng iba pang mga blockchains na nakabatay sa sharding na nangangailangan ng PoW upang pumili ng mga validator, ang Harmony ay batay sa PoS at kaya ang enerhiya ay matipid. Naabot ang pinagkasunduan sa isang linearly scalable algorithm ng BFT na 100 beses na mas mabilis kaysa sa PBFT.
- **Adaptive-Thresholded PoS:** Ang threshold ng stakes na kinakailangan para sa isang node na sumali sa network ay naaakma batay sa dami ng kabuuang staking sa isang paraan na ang mga may masasamang hangarin na stakers ay hindi makapag-concentrate ng kanilang kakayahan sa isang solong shard. Bukod dito, ang threshold ay sapat na mababa upang ang mga maliit na stakers ay maaari pa ring lumahok sa network at kumita ng mga gantimpala.
- **Scalable Networking Infrastructure:** Sa RaptorQ fountain code, ang Harmony ay maaaring magpalaganap ng mga block ng mabilis sa loob ng shards o sa buong network sa pamamagitan ng paggamit ng Adaptive Information Dispersal Algorithm. Ang Harmony ay gumagamit din ng routing ng Kademlia [37] upang makamit ang mga transaksyon sa cross-shard na nasusukat logarithmically kasama ang bilang ng mga shard.
- **Pare-parehong Transaksyon ng Cross-Shard:** Sinusuportahan ng Harmony ang

transaksyong cross-shard na ang mga shard na direktang nakikipag-ugnayan sa bawat isa. Ang isang mekanismo ng atomic locking ay ginagamit upang matiyak ang pagkakapare-pareho ng mga transaksyon sa cross-shard.

Sa pamamagitan ng pagpapabago sa parehong protocol at network layer, ang Harmony ay nagbibigay ng mundo na may scalable at secure blockchain system na maaaring suportahan ang umuusbong na desentralisadong ekonomiya. Ang Harmony ay magbibigay-daan sa mga application na hindi pa nagagawa sa blockchain, kabilang ang mga high-volume na desentralisadong palitan, interactive fair games, mga sistema ng pagbabayad sa Visa, at mga transaksyon sa Internet-of-Things. Nagsusumikap ang Harmony upang mapagkatiwalaan ng bilyun-bilyong tao at lumikha ng radikal na patas na ekonomiya.

2. Mekanismo ng Pinagkasunduan

Ang pinagkasunduan na protocol ay isang mahalagang bahagi ng anumang blockchain. Tinutukoy nito kung gaano kaligtas at kabilis ang mga validator[i] ng blockchain para maabot ang pinagkasunduan sa susunod na block. Ang unang blockchain protocol na pinagkaisahan na nagpapalakas ng Bitcoin ay ang pinagkasunduan na Proof-of-Work (PoW). Ang PoW ay isang proseso kung saan ang mga minero ay naguunahan upang mahanap ang solusyon sa isang cryptographic puzzle-ang nagwagi ay makakakuha ng karapatan na imungkahi ang susunod na block at kumita ng ilang mga gantimpala na token. Ang palagay sa kaligtasan ng PoW ay higit sa 50% ng hashing power na kontrolado ng mga honest node. Sa gayong palagay, ang panuntunan para sa pinagkasunduan na ang pinakamahabang chain ay ang canonical one, at sa gayon ang pinagkasunduang PoW ay tinatawag ding *chain-based consensus* .

Ang iba pang uri ng protocol na pinagkasunduan, ay ang isa na sinaliksik pa sa higit na dalawang dekada sa academia, ay tinatawag na PBFT (Practical Byzantine Fault Tolerance) [14]. Sa PBFT, ang isang node ay inihalal bilang "leader," habang ang natitirang mga node ay "validators." Ang bawat pag-ikot ng pinagkasunduan sa PBFT ay nagsasangkot sa dalawang pangunahing mga yugto: ang yugto ng paghahanda at ang yugto ng paggawa. Sa yugto ng paghahanda, ipina-kalat ng lider ang panukala nito sa lahat ng mga validator, na nagsusumite ng kanilang mga boto ng panukala sa iba. Ang dahilan ng rebroadcasting sa lahat ng mga validator ay ang mga boto ng bawat validator ay kailangang mabilang ng lahat ng iba pang mga validator. Ang paghahanda na bahagi ay natatapos kapag higit sa $2f + 1$ pare-parehong mga boto ang nakikita, kung saan ang f ay ang bilang ng mga may masamang hangarin na validator, at ang kabuuang bilang ng mga validator kasama ang pinuno ay $3f + 1$. Ang yugto ng paggawa ay nagsasangkot ng katulad na proseso ng pagbilang ng boto, at ang pinagkasunduan ay

[i] Ang mga makina na sumusuporta sa blockchain network sa pamamagitan ng pagpapatunay ng mga transaksyon at pag-abot sa pinagkasunduan.

naabot kapag ang mga $2f + 1$ pare-parehong mga boto ay nakikita. Dahil sa rebroadcasting ng mga boto sa mga validator, ang PBFT ay may $O(N^2)$ na pagiging kumplikado sa komunikasyon, na hindi masukat sa isang sistema ng blockchain na may daan-daan o libu-libong node.

Bilang pagpapabuti sa PBFT [14], ang protocol na pinagkasunduan ng Harmony ay linearly scalable sa mga tuntunin ng pagiging kumplikado sa komunikasyon, at sa gayon tinatawag namin itong Fast Byzantine Fault Tolerance (FBFT). Sa FBFT, sa halip na hilingin ang lahat ng mga validator na i-kalat ang kanilang mga boto, ang pinuno ay nagpapatakbo ng isang proseso ng pag-lagda ng multi-signature upang kolektahin ang mga boto ng mga validator sa isang $O(1)$sized multi-signature at pagkatapos ay i-kalat ito. Kaya sa halip na makatanggap ng mga lagda ng $O(N)$, ang bawat validator ay tumatanggap lamang ng isang multi-signature, kaya binabawasan nito ang pagiging kumplikado ng komunikasyon mula sa $O(N^2)$ hanggang sa $O(N)$.

Ang ideya ng paggamit ng $O(1)$-sized multi-signature ay inspired sa ByzCoin's BFT [15] na gumagamit ng Schnorr signature scheme para sa constant-sized multi-signature aggregation at bumubuo ng multicast tree sa mga validator para pangasiwaan ang paghahatid ng mensahe. Gayunpaman, ang isang Schnorr multi-signature ay nangangailangan ng secret commitment round, na humahantong sa kabuuang dalawang round-trip para sa isang solong multi-signature. Ang Harmony ay nagpapabuti sa pamamagitan ng paggamit ng BLS (Boneh-Lynn-Shacham) na multi-signature [28], na nangangailangan lamang ng isang round-trip. Samakatuwid, ang FBFT ay hindi bababa sa 50% na mas mabilis kaysa sa ByzCoin's BFT. Bukod dito, ang Harmony ay gumagamit ng RaptorQ fountain code upang pabilisin ang proseso ng pagkalat ng block (tinalakay sa §6.2). Ang pamamaraan ng pagkalat ng fountain code ay nag-iwas din sa isang isyu sa seguridad sa original tree-based multicasting design ng ByzCoin [16,17].

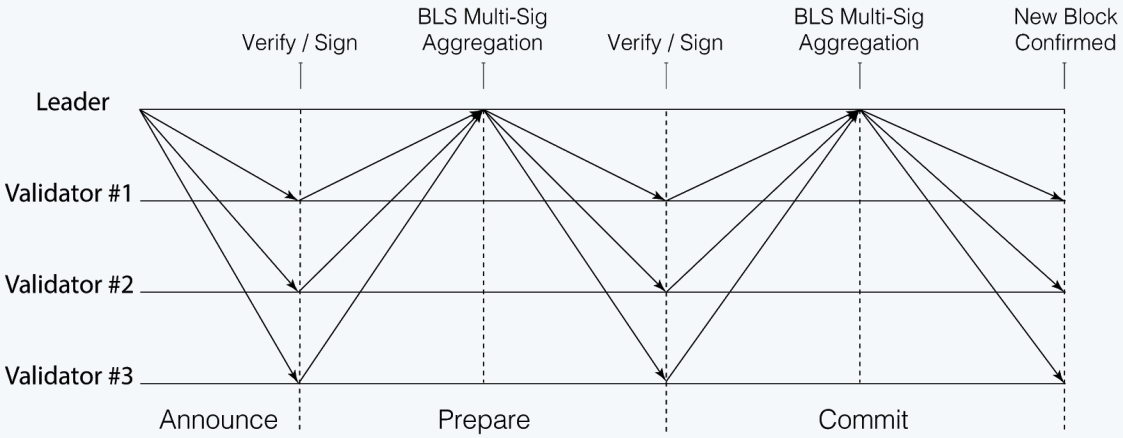

Figure 1. Komunikasyon sa network ng isang solong kasunduan.

Sa partikular, ang kasunduan ng Harmony sa FBFT ay nagsasangkot sa mga sumusunod na hakbang:

1. Ang leader ay nagtatayo ng bagong block at kinakalat ang block header sa lahat ng mga

validator. Samantala, kinakalat ng leader ang nilalaman ng block na may erasure coding (ang mga detalye ay tinalakay sa §6.2). Ito ay tinatawag na "ipahayag" na bahagi.

2. Sinusuri ng mga validator ang bisa ng block header, lagdaan ang block header na may lagda ng BLS, at ipadala ang lagda pabalik sa leader.
3. Ang leader ay maghihintay ng hindi bababa sa $2f + 1$ wastong mga lagda mula sa mga validator (kabilang ang mismong leader) at pinagsasama ang mga ito sa isang BLS multi-signature. Pagkatapos ay ikakalat ng leader ang pinagsama-samang multi-signature kasama ang isang bitmap na nagpapahiwatig kung aling mga validator ang nakapag-sign. Kasama ng Hakbang 2, tinatapos nito ang "maghanda" na bahagi ng PBFT.
4. Sinusuri ng mga validator ang multi-signature kung ito ba ay may hindi bababa sa $2f +$ 1na may lagda, patutunayan ang mga transaksyon sa block content na ikinalat galing sa leader mula sa Hakbang 1, lagdaan ang natanggap na mensahe mula sa Hakbang 3, at ipadala ito pabalik sa leader.
5. Ang leader ay maghihintay ng hindi bababa sa $2f + 1$ na wastong mga lagda (maaaring naiiba ang mga may lagda mula sa Hakbang 3) mula sa Hakbang 4, pinagsasama-sama ang mga ito sa isang BLS multi-signature, at lumilikha ng isang bitmap na logging sa lahat ng mga may lagda. Sa wakas, ang leader ang gumawa ng bagong block sa lahat ng mga multi-signature at bitmaps nakalakip, at ikinalat ang bagong block para sa lahat ng validators na gumawa. Kasama ng Hakbang 4, tinatapos nito ang "gumawa" na bahagi ng PBFT.

Ang mga validator na pinagkasunduan ng Harmony ay inihalal batay sa Proof-of-Stake. Samakatuwid, ang aktwal na protocol ay naiiba nang bahagya mula sa inilarawan sa itaas na ang isang validator na may mas maraming pagbabahagi ng boto ay may higit pang mga boto kaysa sa iba, sa halip na isang-lagda-isang-boto. Kaya sa halip na maghintay ng hindi bababa sa $2f + 1$ na lagda mula sa mga validator, maghihintay ang leader ng mga lagda mula sa mga validator na sama-samang nagtataglay ng hindi bababa sa $2f +$ 1voting shares. Ang mga detalye ng mekanismo ng halalan ng proof-of-stake ay tatalakayin sa §3.3.

3. Sharding

Ang blockchain sharding bilang isang scalability solution ay nakakuha ng maraming atensyon mula noong huling 2017. Iba't-ibang mga solusyon sa sharding ang ipinanukalang pareho sa industriya at academia.

Sa industriya, si Zilliqa [12] ang unang sharding-based public blockchain na nag-claim ng throughput ng

2,800 TPS. Ang Zilliqa ay gumagamit ng PoW bilang proseso ng pagpaparehistro ng pagkakakilanlan (i.e. Sybil attack [1] prevention). Ang network ng Zilliqa ay naglalaman ng isang komite sa directory-service at multiple shard committee (i.e. *network sharding*), bawat isa ay naglalaman ng daan-daang mga node. Ang mga transaksyon ay itinalaga sa iba't ibang mga shard at pinoproseso nang hiwalay (i.e. *transaction sharding*). Ang mga nagresultang block mula sa lahat ng shards ay nakolekta at ipinagsama sa komite ng directory-service. Ang Zilliqa ay hindi isang *state sharding* solution sapagkat ang bawat node ay kailangang hawakan ang buong estado ng blockchain upang maproseso ang mga transaksyon.

Sa akademya, ang mga pahayagan tulad ng Omniledger [8] at RapidChain [7] ay nagpanukala ng mga solusyon na nagtatampok ng *state sharding* kung saan ang bawat shard ay mayroong isang subset ng blockchain state. Ang Omniledger ay gumagamit ng isang pamamaraan na multi-party computation na tinatawag na RandHound [25] upang bumuo ng isang secure random number, na ginagamit upang sapalarang magtalaga ng mga node sa shards. Ipinagpapalagay ng Omniledger ang *slowly adaptive* corruption model kung saan ang mga attackers ay maaaring masira ang lumalaking bahagi ng mga node sa isang shard sa paglipas ng panahon. Sa ilalim ng gayong modelo ng seguridad, ang isang solong shard ay maaaring masira kalaunan. Pinipigilan ng Omniledger ang pagkasira ng mga shard sa pamamagitan ng pagbago ng ayos ng lahat ng mga node sa shards sa isang nakapirming oras na pagitan na tinatawag na *epoch*. Ang RapidChain ay gagawa sa ibabaw ng Omniledger at nagmumungkahi ng paggamit ng *Bounded Cuckoo Rule* upang muling baguhin ang mga node nang walang mga pagkagambala [19].

Ang Harmony ay nakakuha ng inspirasyon mula sa tatlong nakaraang mga solusyon [7,8,12] at nagdidisenyo ng isang ganap na sharding scheme na batay sa PoS na linearly scalable at provably secure. Ang Harmony ay naglalaman ng beacon chain at multiple shard chains. Ang beacon chain ay nagsisilbi bilang randomness beacon at rehistro ng pagkakakilanlan, habang ang shard chains ay nagreserba ng hiwalay na blockchain state at nagpoproseso ng mga kasabay na transaksyon. Ang Harmony ay nagmumungkahi ng mahusay na algorithm para sa randomness generation sa pamamagitan ng pagsasama ng Verifiable Random Function (VRF) at Verifiable Delay Function (VDF). Isinasama din ng Harmony ang PoS sa proseso ng sharding na nagbabago sa pagsasaalang-alang sa seguridad ng isang shard mula sa pinakamaliit na bilang ng mga node [7,8,12] hanggang sa minimum na bilang ng mga pagbabahagi ng boto.

3.1 Distributed Randomness Generation

Background

Ang iba't ibang mga diskarte ay iminungkahi upang magtalaga ng mga node sa mga shard tulad ng randomness-based na sharding [7, 8], location-based sharding [34], at centrally-controlled sharding[35]. Mula sa lahat ng mga diskarte, ang randomness-based na sharding ay kinilala bilang pinaka-secure na solusyon. Sa randomness-based sharding, isang kapwa sumang-ayon sa random na numero ay ginagamit upang matukoy ang sharding assignment para sa bawat node. Ang random na numero ay dapat magkaroon ng mga sumusunod na katangian:

1. Unpredictable: Walang tao ang dapat mahulaan ang random na numero bago ito nalikha.
2. Unbiaseable: Ang proseso ng pagbuo ng random na numero ay hindi dapat maging biasable sa sinumang kasali.
3. Verifiable: Ang bisa ng nabuong random na numero ay dapat mapapatunayan ng sino mang tagamasid.
4. Scalable: Ang algorithm ng randomness generation ay dapat nakasukat sa laki ng bilang ng mga kalahok.

Ang Omniledger [8] ay gumagamit ng RandHound [25] protocol, na isang proseso na leader-driven distributed randomness generation (DRG) na nagsasangkot ng PVSS (Publicly Verifiable Secret Sharing) at Byzantine Agreement. Ang RandHound ay isang $O(n * c^2)$ na protocol na naghihiwalay sa mga node ng kalahok sa maraming grupo ng sukat c. Nakamit nito ang unang tatlong ari-arian sa itaas ngunit impraktikal na mabagal upang maging karapat-dapat bilang scalable.

Ang RapidChain [7] ay tumatagal ng mas simple na diskarte sa pamamagitan ng pagpapaalam sa bawat kalahok na magsagawa ng VSS (Verifiable Secret Sharing) [22] at paggamit ng pinagsamang lihim na pagbabahagi bilang resulting randomness. Sa kasamaang palad, ang protocol na ito ay hindi ligtas dahil ang mga malisosyo na node ay maaaring magpadala ng hindi pantay na pagbabahagi sa iba't ibang mga node [25]. Bukod dito, ang RapidChain ay hindi naglalarawan kung paano naaabot ng mga node ang pinagkasunduan sa maramihang possible version ng reconstructed randomness.

Bilang karagdagan, ang Algorand [18] ay nakasalalay sa VRF-based (Verifiable Random Function) cryptographic na pag-uuri upang piliin ang grupo ng mga validator ng consensus. Ang disenyo ng Ethereum 2.0 ay nagmumungkahi ng paggamit ng VDF (Verifiable Delay Function) [20] upang maantala ang pagbubunyag ng aktwal na random na numero upang maiwasan ang atake ng panghuling tagapagsalita [36]. Ang VDF ay isang bagong likha ng cryptographic primitive; kailangan madaling iakma ang minimum na dami ng oras upang

makalkula at ang resulta ay maaaring ma-verify kaagad.

Scalable Randomness Generation with VRF and VDF

Pinagsasama ng diskarte ng Harmony ang mga lakas ng mga solusyon sa itaas. Una, ang komplikadong protocol na DRG ng Harmony ay $O(n)$, na sa pagsasanay ay hindi bababa sa isang order ng tindi na mas mabilis kaysa sa RandHound. Pangalawa, hindi katulad sa simpleng diskarte ng RapidChain na VSS-based, ang amin ay unbiasable at napapatunayan. Ikatlo, kumpara sa solusyon ng Ethereum 2.0, ang aming diskarte ay gumagamit ng kasunduan ng BFT upang magbigay ng kawakasan sa random na numero. Sa partikular, ang protocol ay kinabibilangan ng mga sumusunod na hakbang:

1. Ang leader ay nagpapadala ng *init* message sa hash ng huling block $H(B_{n-1})$ sa lahat ng mga validator.
2. Para sa bawat validator i, matapos matanggap ang *init* message, ang isang VRF ay kinikwenta upang lumikha ng isang random na numero r_i at isang patunay p_i: $(r_i, p_i) = VRF(sk_i, H(B_{n-1}))$, kung saan ang sk_i ay ang sikretong key ng validator i at v ay ang kasalukuyang numero ng view ng pinagkasunduan. Pagkatapos, ang bawat validator ay nagpapadala babalik (r_i, p_i) sa pinuno.
3. Maghihintay ang lider hanggang sa makatanggap ito ng hindi bababa sa $f + 1$ na wastong random na mga numero at pinagsasama ang mga ito sa XOR operation upang makuha ang preimage ng panghuling randomness na $pRnd$.
4. Ang leader ay nagpapatakbo ng BFT (tinalakay sa §2) sa lahat ng mga validator upang maabot ang pinagkasunduan sa $pRnd$ at ipagtapat ito sa block B_n.
5. Pagkatapos ng $pRnd$ ay nakatuon, ang leader ay magsisimulang kuhain ang aktwal na randomness $Rnd = VDF(pRnd, T)$, kung saan ang T ay ang kahirapan ng VDF at itinakda sa algorithm tulad na ang randomness ay maaari lamang kuhain pagkatapos ng k blocks.
6. Sa sandaling makuha ang Rnd, ang pinuno ay magsisimula ng isang BFT sa lahat ng mga validator upang sumang-ayon sa bisa ng Rnd at sa wakas ay magkakaroon ng randomness sa blockchain.

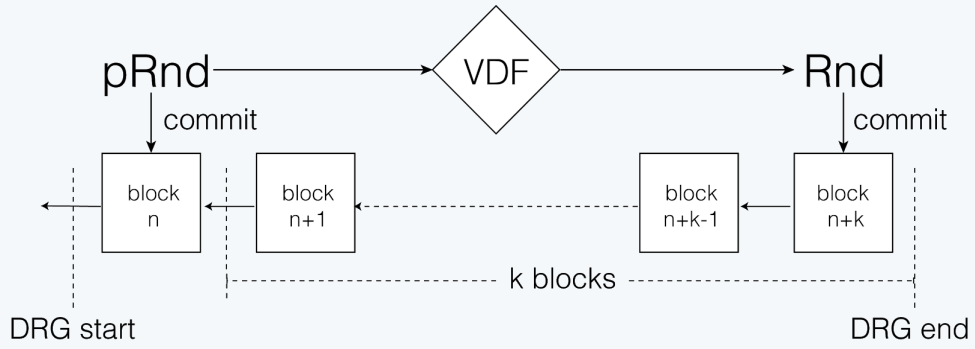

Figure 2. Ang VDF (Verifiable Delay Function) ay inaantala ang pagbubunyag ng panghuling randomness.

Ang VDF ay ginagamit upang mapaglabanan ang pagpapahayag ng Rnd at maiwasan ang isang malisosyong leader mula sa pagbubuklod ng randomness sa pamamagitan ng cherry-picking ng subset ng mga random na numero ng VRF. Dahil sa VDF, ang leader ay hindi maaaring malaman ang aktwal na panghuling randomness bago ang $pRnd$ ay nakatuon sa blockchain. Sa oras na ang Rnd ay nakalkula sa VDF, ang $pRnd$ ay nakatuon na sa nakaraang block upang ang leader ay hindi maaaring manipulahin ito ngayon. Samakatuwid, ang pinakamahusay na isang malisosyong leader ay maaaring gawin ang alinman sa randomness $pRnd$, o pwestuhan ang protocol sa pamamagitan ng hindi paggawa sa $pRnd$. Ang nakaraan ay katulad ng honest behavior. Ang huli ay hindi magiging sanhi ng maraming pinsala dahil gaya ng parehong timeout mechaniam sa PBFT [14] ay gagamitin upang palitan ang leader at simulan muli ang protocol.

Ipinapalagay namin, sa katagalan, ang pag-iral ng ASICs upang makalkula ang VDFs, kung saan ang ilang mga altruistic node na tumatakbo sa isang ASIC (Application-Specific Integrated Circuit) ay mag-publish ng resulta, at walang sinuman ang makakapaglaro sa sistema. Ito ay posible, bago isagawa ang VDF ASICs, na ang isang magsasalakay na may isang mas mabilis na computing device ay maaaring makalkula ang resulta bago ang iba pang honest node. Hanggang sa mangyari ito, ang mga attacker lamang ang maaring makaalam ng bahagyang randomness bago ang honest nodes. Habang sa principle sino man magtangkang sumalakay ay maaaring samantalahin ito (e.g. pag-withdraw ng pondo nito kung ang taya sa smart contract ay hindi pabor sa kanya), ang problemang ito ay maaaring mapigilan sa smart contract layer na may proper delay katulad ng dapat na paghihintay sa panahon para sa randomness na nakatuon sa protocol bago ang isang withdrawal ng pondo ay maging posible.

3.2 Epochs

Sa Harmony, ang napagkasunduan at pang-sharding na proseso ay orchestrated sa pamamagitan ng konsepto ng epochs. Ang epoch ay isang predetermined time interval (e.g. 24 oras) na panahon kung saan ang istraktura ng sharding ay maayos at ang bawat shard ay patuloy na nagpapatakbo ng pinagkasunduan sa parehong hanay ng mga validator. Sa simula ng bawat epoch, ang isang random na numero ay bubuuin gamit ang DRG protocol na inilarawan sa §3.1, at ang istraktura ng sharding ay matutukoy batay sa randomness. Ang mga validator na gustong patunayan ang mga transaksyon sa epoch e ay kailangang istake ang kanilang mga token sa panahon $e - 1$. Ang cutoff na oras para sa staking ay bago ang randomness preimage $pRnd$ ay nakatuon sa blockchain.

3.3 Staking-based Sharding

Pagpaparehistro ng Validator

Ang pag-atake ng Sybil [1] ay isang pangunahing pagsasaalang-alang sa seguridad ng mga pampublikong blockchain. Kinakailangan ng Bitcoin at Ethereum ang mga minero na mag-compute ng isang cryptographic puzzle (PoW) bago sila magpanukala ng isang block. Gayundin, ang mga blockchain na batay sa sharding tulad ng Zilliqa [12] o Quarkchain [11] ay gumagamit din ng PoW upang maiwasan ang pag-atake ng Sybil. Ang Harmony ay gumagamit ng iba't ibang diskarte sa proof-of-stake (PoS) bilang pagpaparehistro ng validator o mekanismo na pagpigil sa pag-atake ng Sybil. Upang maging isang validator ng Harmony, ang mga prospective na kalahok (o stakers) ay dapat magstake ng isang tiyak na halaga ng mga token upang maging karapat-dapat. Ang bilang ng mga token na nakastake ay matutukoy ang bilang ng voting shares na nakatalaga sa validator. Ang bawat voting share ay tumutugma sa isang boto sa pinagkasunduan ng BFT (tulad ng tinalakay sa §2).

Sharding ng Voting Shares

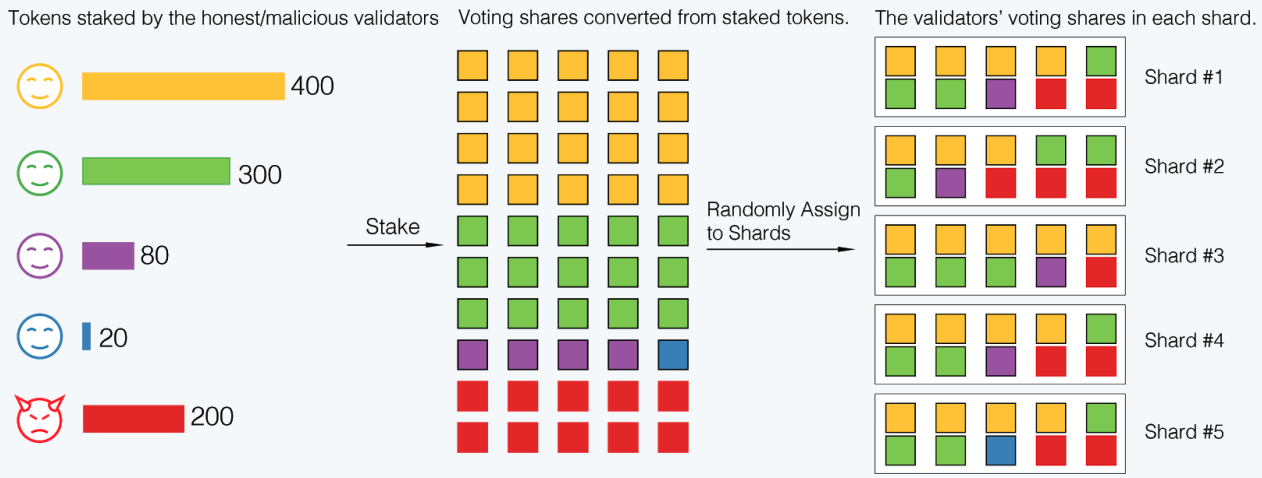

Figure 3. Ang mga stakers ay nakuha ang voting shares ng proporsyonal sa kanilang mga staked token. Ang voting shares ay random na itatalaga sa shards. Ang mga stakers ay magiging mga validator para sa mga shard kung saan nakatalaga ang kanilang voting shares.

Ang voting share ay isang virtual ticket na nagpapahintulot sa isang validator na magsumite ng isang boto sa pinagkasunduan. Maaaring makuha ng mga validator ang voting shares sa pamamagitan ng staking tokens. Ang bilang ng mga token na kinakailangan para sa isang voting share ay naaayos ng algorithm. Sa simula ng bawat epoch, ang voting shares ng mga bagong validator ay random na nakatalaga sa shards. Ang mga bagong validator ay sumali sa mga shard (s) kung saan ang kanilang voting shares ay nakatalaga. Tulad ng tinalakay sa §2, ang pinagkasunduan sa isang shard ay naabot ng mga validator na sama-samang nagtataglay ng hindi bababa sa $2f + 1$ na voting shares upang lagdaan ang block.

Upang garantiyahan ang seguridad ng isang single shard, ang bilang ng voting shares sa mga malisosyong validator ay kailangang itago pababa $\frac{1}{3}$ ng lahat ng voting shares sa shard na iyon. Ito ay kinakailangan dahil sa likas na katangian ng kasunduan ng BFT. Ginagarantiyahan ng Harmony's thresholded PoS ang mga kinakailangan sa seguridad sa itaas sa pamamagitan ng adaptive na pagsasaayos ng presyo ng voting share at pagtatalaga ng indibidwal na voting share sa mga shard kaysa sa mga indibidwal na validator.

Ang aming palagay sa seguridad ay nasa kabuuan ng lahat ng staked token, hanggang sa $\frac{1}{4}$ kanila ang nabibilang sa malisosyong validator. Kung ishard namin sa pamamagitan ng mga validator (i.e. magtalaga ng isang validator sa isang shard), sa pinakamasamang kaso kung saan ang isang solong validator ay nakahadlang na mayroong $\frac{1}{4}$ ng lahat ng mga staked token (o ang voting shares), madali lang itong magkaroon ng higit sa $\frac{1}{3}$ na voting share sa shard na iyon . Ang dahilan dito ay ang mga stake sa bawat shard ay merong m beses na mas mababa kaysa sa mga stake ng buong network, kung saan ang m ay ang bilang ng shards. Tinatawag namin Ang sitwasyon ng pag atake na *large-stake attack* (isang espesyal na uri ng pag-atake sa single-shard).

Upang maiwasan ang pag-atake ng *large-stake*, sa halip na mag-shard sa pamamagitan ng mga validator, nagshashard kami sa pamamagitan ng voting shares (i.e. magtalaga ng isang voting share sa isang shard). Sa partikular, pagkatapos na maihayag ang Rnd sa simula ng kasalukuyang epoch, ang isang random na permutasyon (seeded na may Rnd) sa lahat ng voting shares ay tapos na at ang pinapahintulutang listahan ng voting shares ay magkakahati sa m bucket, kung saan ang m ay bilang ng shards. Ang voting shares ay bumabagsak sa ith bucket na itinalaga sa shard i, gayon din ang katumbas na mga validator. Sa pagsasagawa, ang isang solong validator ay maaaring italaga sa maraming shards kung mayroon siyang voting shares na nakatalaga sa mga shards na iyon. Ang leader ng shard ay natutukoy bilang mga validator na nagtataglay ng unang voting share sa bucket.

Mahalagang tandaan na ang mga validator na may mas malaking stakes ay magkakaroon ng mas maraming pagkakataon na mapili bilang leader. Nagtalo kami na talagang ito ay isang kanais-nais na sitwasyon dahil ang mga malalaking stakers ay may mas maraming insentibo na sundin ang protocol dahil sa takot sa kanilang stake na maslash (ang mekanismo ng insentibo ay tinalakay sa §7). Bilang karagdagan, ang mga ito ay mas malamang na magkaroon ng mas makapangyarihang mga machine na may mabilis at matatag na network.

Adaptive-Thresholded PoS

Ang presyo ng voting share ay naka-handa pa algorithmically upang ito ay maging sapat na mapaliit upang ang mga malicious stakers ay hindi maaaring tumutok sa kanilang kapangyarihan pagboto sa isang solong shard. Sa partikular, itinakda namin ang presyo ng isang voting share upang maging P_{vote}: token:

$$P_{vote} = \overline{NumTSShea\text{-}r_1d} * \lambda$$

Ditoλ ang isang security parameter, $NumShard$ ay ang numero ng shards at TS_{e-1}ay ang kabuuang bilang ng mga token na staked sa habang epoch $e - 1$.

Ngayon ay pinapatunayan namin na kung kailan $\lambda > 600$, ang pagkakataon ng isang shard ay magkakaroon ng mas maraming$\frac{1}{3}$ malisosyong voting shares (i.e.posibilidad ng kabiguan) ay bale-wala.

Ibinigay ang kahulugan ngP_{vote}, ang kabuuang bilang ng voting shares ay magiging $N = T_{P_{vote}} = NumShard * \lambda$.

Dahil sa isang pinagkakatiwalaan na pinagmulan ng ramdomness (tinalakay sa §3.1) at ang proseso ng sharding ay nakabase sa randomness,ang probabilidad ng pamamahagi ng numero ng malisosyong voting share sa bawat shard ay maaring maimodelo bilang isang hypergeometric na pamamahagi (i.e. random sampling na walang kapalit):

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

Narito N ay ang kabuuang bilang ng voting shares, $K = \frac{N}{4}$ay ang pinakamataas na bilang ng malisosyong voting shares, $n = \frac{N}{Num_{Shard}}$ ang bilang ng voting shares sa bawat isang shard,at k ang bilang ng malisosyong voting shares sa isang shard. Ang aktwal na kabiguan sa paggagrado ng shard $P(X \leq k)$ay sumusunod sa pinagsamang hypergeometric na pamamahagi

$CDF_{hg}(N, K, n, k)$ kung saan, kapag ang N ay malaki, degrades sa binomial na pamamahagi (i.e. random sampling na may kapalit):

$$P(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$$

Maaari naming ipakita na kapag n ay may sapat na laki, ang posibilidad na ang isang shard ay naglalaman ng higit sa $\frac{1}{3}$ na mga token na hawak ng mga malicious entities na hindi mababawasan. Sa katunayan, kapag$n$ = 600, ang posibilidad na ang isang shard ay naglalaman ng hindi bababa sa $\frac{1}{3}$ malicious voting shares ay$P(X \leq 200)$ = 0.999997, na isinasalin sa isang kabiguan ng shard (i.e. hindi naabot ang pinagkasunduan) rate ng “isang beses sa loob ng isang libong taon” (binigyan ng agwat na epoch ng 24 oras). Samakatuwid, itatakda namin ang λ = 600 upang magarantiya ang mataas na seguridad ng aming mga shard. (Intuitively, λay nangasiwa ng minimum na bilang ng voting shares ng isang solong shard na dapat ay nilalaman. Ito ay gumagana katulad sa minimum na bilang ng mga node sa isang shard tulad ng inilarawan sa iba pang mga solusyon na batay sa PoW sharding [7,8,12])

Ang diskarte na ito ay resistant sa pagbabago bago ng mga bilang ng mga validator. Hindi kami nagtatakda ng mas mababang limitasyon sa bilang ng mga validator sa bawat shard tulad ng sa iba pang mga solusyon tulad ng Zilliqa [12]. Sa halip, naglagay kami ng isang adaptive na PoS-based na modelo upang matiyak na ang mga malisosyong na tao ay hindi maaaring maghawak ng higit sa $\frac{1}{3}$ng voting shares sa isang solong shard, kaya ginagawa itong ligtas.

3.4 Resharding

Inilarawan namin ang isang ligtas na pamamaraan ng sharding na pumipigil sa mga malisosyong validator na makakuha maski isang shard. Gayunpaman, kung ang istraktura ng sharding ay mananatiling maayos, ang malisosyong attackers ay maaari pa ring makakuha ng shard sa pamamagitan ng pag-corrupt sa mga validator sa shard na iyon. Mayroong tatlong modelo ng attackers:

1. Static Round-Adaptive: kung saan ang mga attackers ay maaari lamang makasira ng isang subset ng mga node sa isang paunang natukoy na yugto. Ang Elastico [9] ay nagpapahiwatig na maaari lamang ang attacker ay makasira ng mga node sa simula ng bawat epoch.
2. Slowly Adaptive: kung saan ang mga attackers ay maaari lamang makasira ng isang subset ng mga nodes sa matagal na panahon habang ang epoch[7,8].
3. Fully Adaptive: kung saan ang mga attackers ay maaari lamang makasira ng isang subset ng mga nodes agad at anumang oras mula ngayon[18].

Ipinagpapalagay ng Harmony ang mabagal na adaptive corruption model na kung saan maaaring masira ng attacker ang isang pare-parehong bilang ng mga node at kailangan muna nila ng sapat na oras. Ipinagpapalagay ng Omniledger [8] ang parehong corruption model at pinipigilan nito ang mga pagatake sa pamamagitan ng pagpapalit ng mga validator sa lahat ng shards sa bawat epoch. Ang diskarte na ito ay may dalawang pangunahing problema. Ang una ay ang mataas na halaga ng bootstrapping sa bawat epoch. Ang ikalawa ay ang pag-alala sa seguridad na kapag ang lahat ng mga node ay pinalitan sa panahon ng pinagkasunduan.

Ang Harmony ay gagawa ng paraan para malunasan ang mga problemang ito sa pamamagitan ng pagadopt sa Cuckoo-rule based resharding mechanism [7,19]. Matapos ang katapusan ng isang epoch, ang mga validator na kinuha ang kanilang stake ay matatanggal na mula sa network, habang ang ibang mga pinanatili ang kanilang mga stakes ay mananatili. Ang mga bagong validator na may stake sa panahon ng epoch ay nakakakuha ng bagong voting shares. Ang voting shares ay random na nakatalaga sa shards na may higit sa median ng kabuuang voting shares. Susunod, ang isang pare-pareho na bilang ng voting shares mula sa lahat ng shards ay random na ibabahaging muli sa kabilang kalahati ng shards na mas mababa kaysa sa median ng kabuuang voting shares. Ito ay napatunayan [7] na ang resharding scheme na ito ay maaaring panatilihin ang voting shares sa lahat ng mga shards na balanse habang tinutupad ang seguridad na kailangan.

3.5 Fast State Synchronization

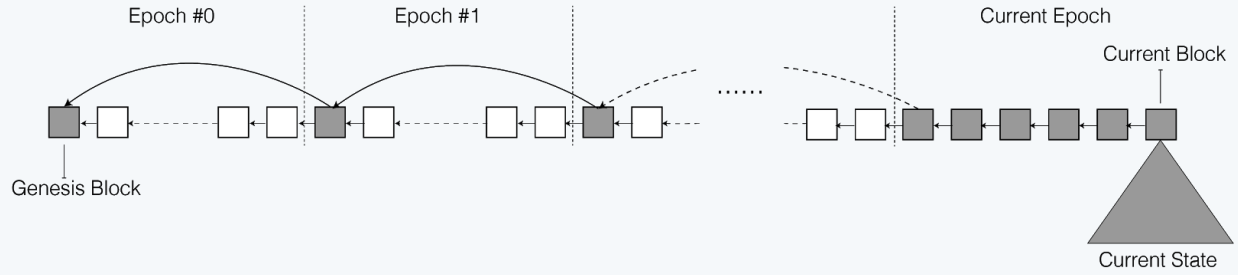

Figure 4. Ang unang block ng isang epoch ay naglalaman ng isang hash link sa unang block ng huling epoch. Pinapayagan nito ang mabilis na state synchronization ng mga bagong node kung saan maaari silang dumipende sa mga block na grey upang mabilis na i-verify ang kasalukuyang estado.

Kapag ang validators ay sumali sa isang bagong shard, kailangan nilang mabilis na i-synchronize sa kasalukuyang estado ng shard upang patunayan ang mga bagong transaksyon. Ang tradisyunal na pamamaraan ng pag-download ng blockchain history at muling pagtatayo ng kasalukuyang estado ay masyadong mabagal para sa resharding, upang maging posible (ito ay tumatagal ng mga ilang araw upang ganap na i-synchronize ang Ethereum blockchain history). Sa kabutihang palad, ang kasalukuyang estado ay may magnitude order lamang na maliit kaysa sa buong blockchain history. Ang pag-download ng kasalukuyang

estado ng epoch ay magagawa kumpara sa pag-download ng buong history.

Sa Harmony, ang mga bagong validator na sumali sa isang shard kailangan nilang unang i-download ang kasalukuyang trie ng estado ng shard na iyon upang maaari nilang simulan ang mabilis na pagpapatunay ng mga transaksyon. Upang matiyak na wasto ang kasalukuyang nai-download na estado, kailangan ng bagong node upang gawin ang wastong pag-verify. Sa halip na i-download ang buong blockchain history at balikan ang lahat ng mga transaksyon upang patunayan ang kasalukuyang estado, ang bagong node ay nagda-download ng mga makasaysayang block header at pinapatunayan ang mga header sa pamamagitan ng pagsuri sa kanilang mga lagda. Hangga't mayroong isang cryptographic trace (e.g.hash pointers at mga lagda) mula sa kasalukuyang estado pabalik sa block ng genesis, ang estado ay may bisa. Gayunpaman, ang pagpapatunay ng mga lagda ay hindi computationally free at nangangailangan ito ng isang mahabang panahon upang patunayan ang lahat ng mga lagda na nagsisimula sa block ng genesis. Upang pagaanin ang problemang ito, ang unang block ng bawat epoch ay magsasama ng karagdagang hash pointer sa unang block ng huling epoch. Sa ganitong paraan, ang bagong node ay maaaring lumipat sa kabilang mga block sa loob ng isang epoch kapag natrace ang hash pointers ng genesis block. Ito ay malaking tulong upang mapabilis ang pagpapatunay ng kasalukuyang blockchain state.

Upang higit pang ma-optimize ang state synchronization process, gagawa kami ng blockchain state na mas maliit hangga't maaari. Sa pagmamasid mula sa kalagayan ng Ethereum blockchain state ay maraming mga account ang walang mga laman at nag-aaksaya ng mahalagang puwang ng blockchain state. Sa Ethereum, ang mga walang laman na account na may isang partikular na nonce ay hindi maaaring matanggal dahil sa mga potensyal na pag-atake kung saan ang mga lumang transaksyon ay maaaring muling isinumite sa tinanggal na account [32]. Ang Harmony ay magpapatupad ng iba't ibang modelo ng pag-iwas sa mga paulit ulit na pag-atake sa pamamagitan ng pagpapaalam sa mga transaksyon na tukuyin ang hash ng kasalukuyang block: ang transaksyon ay may bisa lamang bago ang isang tiyak na numero (e.g. 100) ng mga block na sinusundan ng block na may specified na hash. Sa ganitong paraan, ang mga lumang account ay maaaring maligtas mula sa pagkabura at ang kalagayan ng blockchain ay mapanatiling slim.

4. Shard Chain and Beacon Chain

4.1 Shard Chain

Ang isang shard chain ay isang blockchain na nagpoproseso at nagpapatunay sa sarili nitong mga transaksyon at nagtatago ng sarili nitong state. Ang isang shard ay nagpaproseso lamang ng mga transaksyon na may kaugnayan sa sarili nito. Kahit na ang isang shard chain ay relatibong independent, makikikumunikasyon pa din ito sa iba pang mga chains sa pamamagitan ng cross-shard communication.

Cross-shard Communication

Ang cross-shard communication ay isang mahalagang bahagi ng anumang sharding na batay sa blockhain. Ang kakayahan ng cross-shard ay may kakayahan na sirain ang barrier sa pagitan ng mga shard at nagpapalawak ng utility ng isang solong shard na lampas sa sarili nito.

Sa pangkalahatan, mayroong tatlong kategorya ang cross-shard communication:

1. Main-chain-driven: Ang mga proyekto tulad ng Zilliqa [12] ay umaasa sa pangunahing chain upang makamit ang mga transaksyon sa buong shards.
2. Client-driven: Ang Omniledger [8] ay nagpanukala ng mekanismo sa transaksyon ng cross-shard na client-driven kung saan ang mga mensahe sa pagitan ng mga shards ay nakolekta at ipinadala sa mga shards ng mga client. Nagdadagdag ito ng dagdag na pasanin sa client na hindi kanais-nais para sa isang adhoc light client.
3. Shard-driven: Ipinanukala ng RapidChain [7] na ang mga mensahe sa pagitan ng mga shard ay direktang ipinadadala ng mga node sa shard na walang kailangang tulong.

Ang Harmony ay nagadopt ng shard-driven approach para sa pagiging simple nito at ang kawalan ng pasanin sa mga kliyente. Naniniwala kami na ang mga benepisyo ng komunikasyon sa shard ay mas malaki kaysa sa mga kakulangan nito. Ang gastos sa pangkalahatang network para sa shard-driven na komunikasyon ay maaaring malaki sapagkat ang bawat mensahe ng cross-shard ay isang network-level na naikalat, na kung saan ay nagkakaroon ng gastos sa network ng $O(N)$. Upang malutas ang problemang ito, Ang Harmony ay gumamit ng routing protocol ng Kademlia upang mabawasan ang komplikadong komunikasyon sa $O(log(N))$. Bilang karagdagan, ang data na ipinapadala ay naka-encode na at may erasure code upang matiyak ang katatagan ng komunikasyon ng cross-shard. Ang mga detalye ay tatalakayin sa §6.

4.2 Beacon Chain

Ang Harmony beacon chain ay isang espesyal na blockchain na naghahain ng mga karagdagang layunin kumpara sa mga shard chain. Sa epekto, ang beacon chain ay isang shard din. Bukod sa pagproseso ng mga transaksyon, tulad ng iba pang mga shard chain, ang beacon chain ay namumuno ng dalawang karagdagang key functionality: pagbuo ng random na numero (tinalakay sa §3.1) at pagtanggap ng mga stakes, na nangangahulugang ang beacon chain ay ang chain kung saan itatabi ng mga stakers ang kanilang mga token upang maging mga validator.

Ang mga validator para sa beacon chain ay natutukoy katulad ng iba pang mga shard chain. Sa panahon ng pagtatalaga ng sharding, ang voting shares ay random na nahahati sa $NumShard + b$ buckets, kung saan ang mga extra b buckets ay para sa beacon chain.

Hash Link from Shard Chain

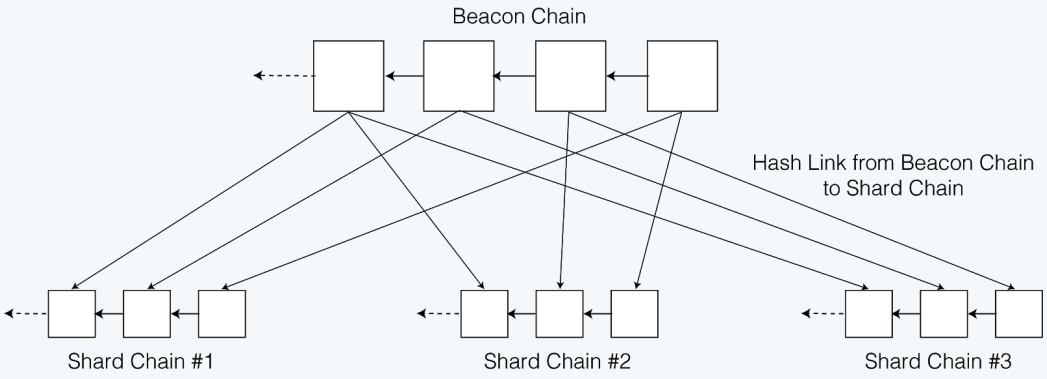

Figure 5. Hash link mula sa beacon chain block papunta sa shard chain block.

Ang beacon chain ay tumutulong na pataasin ang seguridad at pagkakapare-pareho ng mga estado ng shard chain sa pamamagitan ng pagsasama ng block header mula sa bawat shard chain. Sa partikular, pagkatapos ng isang bagong block ba nakatuon sa isang shard chain, ipapadala ang block header nito (inter-shard communication na nakabatay sa Kademlia) sa beacon chain. Sinusuri ng beacon chain ang pagiging wasto ng header ng block sa pamamagitan ng:

1. Ang hash ng kanyang nakaraang bloke, na dapat na nakatuon sa beacon chain;
2. Ang mga signers ng multi-signature ng mga block, na dapat ang tamang validators lamang para sa shard na iyon.

Ang mga nakatalagang mga header ng block sa beacon chain ay ikakalat sa buong network. Ang bawat shard ay magpapanatili ng chain na wasto sa mga header ng block para sa lahat ng iba pang mga shard, na gagamitin upang suriin ang bisa ng mga transaksyon mula sa iba pang

mga shard (i.e. simpleng pagpapatunay ng pagbabayad). Ang pagdaragdag ng mga header ng block ng shard chain sa beacon chain ay nagsisilbi ng dalawang pangunahing layunin:

1. Tumataas ang seguridad ng isang solong shard.
   Kailangan muna icorrupt o atakihin ang parehong shard chain at beacon chain upang kumbinsihin ang iba na ang alternatibong block sa shard chain ay may bisa.
2. Bawasan ang halaga ng network ng pagsasahimpapawid ng mga block header sa mga shard.
   Magkakaroon ng isang komunikasyon sa network ng $O(N^2)$ kung hayaan natin ang bawat shard na ikalat nang hiwalay ang mga header nito. Sa beacon chain bilang sentrong relay, ang pagiging kumplikado ay nabawasan sa $O(N)$.

5. Blockchain State Sharding

Hindi tulad ng ibang mga state-sharding blockchains [7,8] na gumamit ng data model ng UTXO (Unspent Transaction Output), Ang harmony state-sharding ay inilagay sa modelo ng data na nakabatay sa account. Ang bawat shard chain ay naglalaman ng sarili nitong state account, at ang lahat ng token na lumalaganap ay kumakalat sa lahat ng shard.

Tinatrato namin ang user account at ang smart contract account ng naiiba sa sharding. Maaaring magkaroon ng maramihang balanse ang isang user account sa iba't ibang mga shard (e.g. 100 na token sa Shard A at 50 na token sa Shard B). Maaaring ilipat ng isang user account ang balanse nito sa pagitan ng shards sa pamamagitan ng pag-isyu ng isang cross-shard transaction. Ang isang smart contract account ay limitado sa mga tiyak na shard kung saan ang kontrata ay nilikha. Gayunpaman, para sa isang desentralisadong application na nangangailangan ng mas maraming throughput kaysa sa isang solong shard ay maaaring hawakan, ang Dapp (Desentralized Application) na ang developer ay maaaring magbigay ng halimbawa ng maraming mga pagkakataon ng parehong smart contract sa iba't ibang shards at hayaang hawakan ng bawat pagkakataon ang isang subset ng papasok na trapiko. Tandaan na ang iba't ibang mga pagkakataon ng parehong smart contract ay hindi nagbabahagi ng parehong estado, ngunit maaari silang makipag-ugnayan sa isa't isa sa pamamagitan ng komunikasyon sa cross-shard.

6. Networking

Ang nakaraang pananaliksik [33] ay itinuturo na ang kapasidad ng network ay isa sa mga pangunahing bottleneck para sa mga sistema ng blockchain. Upang mas mapataas ang performance, nakatuon ang Harmony sa pagpapabuti ng kahusayan ng paggamit ng network. Ang Harmony ay nagmumungkahi din ng maraming mga pagpapabuti upang harapin ang mga nangyayari sa real-world networking.

6.1 Kademlia-based Routing

May inspirasyon ng RapidChain [7], tatanggapin namin ang Kademlia [37] bilang mekanismo ng pagruruta para sa mga mensahe ng cross-shard. Ang bawat node sa network ng Harmony ay nagpapanatili ng routing table na naglalaman ng mga node mula sa iba't ibang mga shard. Ang distansya sa pagitan ng mga shards ay tinukoy bilang XOR distance ng mga ID ng shard. Kapag ang isang mensahe mula sa shard A ay kailangang ipadala sa shard B, ang mga node sa shard A ay titignan ang routing table at ipadala ang mensahe sa mga node na gamit ang pinakamalapit na shard ID. Sa routing batay sa Kademlia, ang isang mensahe ay naglalakbay lamang sa mga node ng $O(logN)$ bago ito umabot sa patutunguhang shard. Kung ikukumpara sa normal na pagsasahimpapawid ng gossip, na nangangailangan ng kumplekadong network ng $O(N)$, ang mekanismo ng routing ng Kademlia ay maaaring makabuluhang bawasan ang kabuuang load ng network sa isang sharded blockchain.

6.2 Mahusay na Pagkalat na may Erasure Code

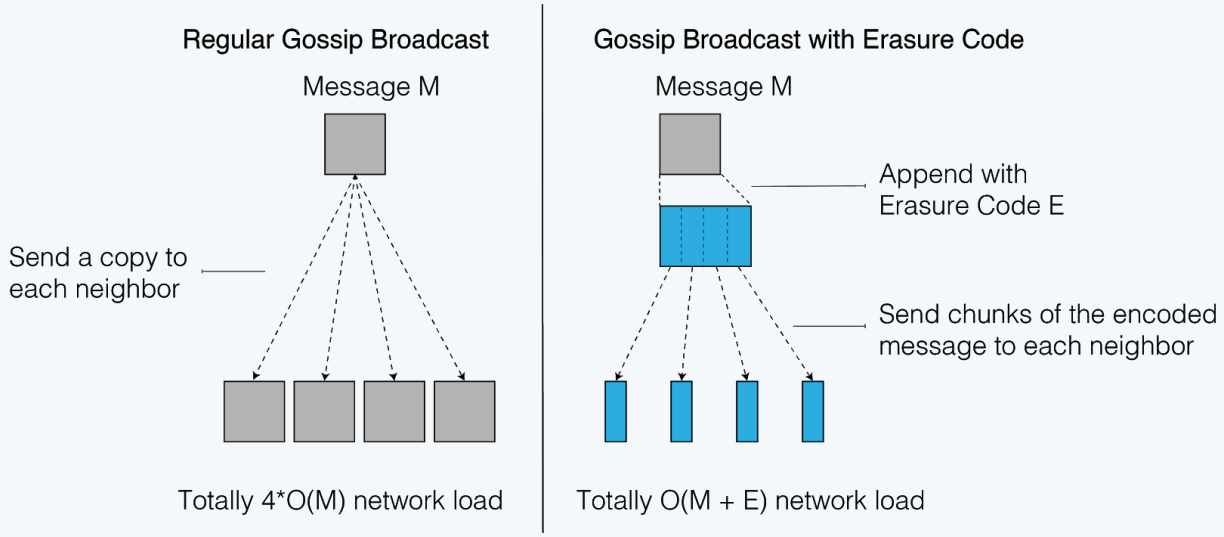

Figure 6. Paghahambing sa pagitan ng normal gossip na broadcast na may gossip na broadcast na may erasure code.

Ang broadcast ay isang madalas na pagkilos ng network sa anumang blockchain na sistema na binuo sa network overley P2P (Peer-to-Peer). Sa partikular sa aming protocol ng

pinagkasunduan, may tatlong mga sitwasyon kung saan kailangan ang pagsasahimpapawid:

1. Ang isang bagong ipinanukalang block ay kailangang ma-ikalat ng pinuno sa lahat ng mga validator.
2. Ang isang bagong binuong master chain block ay kailangang ma-ikalat sa buong network.
3. Ang komunikasyon ng cross shard ay nangangailangan ng pag-kalat ng isang mensahe sa pagitan ng mga shard.

Sa mga normal na pagsasahimpapawid sa P2P, ang orihinal na nagpadala ay kailangang magpadala ng kopya ng mensahe sa bawat isa sa mga neighbor nito. Ito ay magkakaroon ng load ng network ng $O(d * M)$ sa nagpadala, kung saan ang d ay ang average na bilang ng mga neighbor ng nagpadala at M ay ang laki ng mensahe. Sa halip, sa Harmony, ang nagpadala ay unang naka-encode ng mensahe na may erasure code at pagkatapos ay nagpadala ng mga chunk ng naka-encode na mensahe sa bawat neighbor. Binabawasan nito ang load sa nagpadala sa $O(M + e)$ kung saan ang e ay ang laki ng erasure code at karaniwan ito ay mas maliit kaysa sa laki ng orihinal na mensahe M. Samakatuwid, ang mekanismo ng pagsasahimpapawid ng network ng Harmony ay makabuluhang nagpapababa sa network load ng nagpapadala. Bilang karagdagan, ang Harmony ay nagmumungkahi na mapabuti ang katatagan ng IDA sa pamamagitan ng pagpapalit ng orihinal na erasure code ng Reed-Solomon sa fountain code ng RaptorQ upang ang broadcaster ay palaging magpadala ng higit pang mga erasure code upang masiguro na ang datos ay natanggap sa kalaunan.

6.3 FEC-based Unicast

Ang tradisyunal na maaasahang transportasyon tulad ng TCP [42] ay nakasalalay sa retransmission at ACK-based na nagbibigay ng senyas upang makitungo sa mga lost packet. Ito ay kilala upang ipakilala ang mga spike ng latency na proporsyonal sa oras ng pag-ikot sa pagitan ng nagpadala at tagatanggap. Gayundin, ang control ng congestion na window-based-tulad ng Reno, NewReno, and CUBIC na ginagamit ng karamihan sa mga pagpapatupad ng TCP-ay ang lahat ng additive ng algorithm na increase/multiplicative sa pagbaba (AIMD), na ang bandwidth ay kilala na malubhang napinsala ng mga lumilipas na packet losses.

Ginagamit ng Harmony ang fountain code ng RaptorQ upang labanan ang dalawang problemang ito. Ang bawat mensahe ay naka-encode sa mga simbolo, at ang mga simbolo ay ipinapadala sa ibabaw ng wire hanggang tinatanggap ng receiver ang matagumpay na pag-decode ng mensahe gamit ang mga simbolo na natanggap nito. Hindi tulad ng paggamit ng mga nakapirming mga codes tulad ng Reed–Solomon kung saan ang paghahatid ay nabigo sa sandaling ang mga simbolo ay naubos na, ang fountain code ay nagbibigay ng walang

katapusan, makatarungang henerasyon at paggamit ng mga simbolo ng pag-encode.

6.4 Support or Home Nodes

Ang mga P2P node sa isang typical residential network ay nagpapakita ng isang pangunahing, natatanging problema: Hindi sila maaabot mula sa labas maliban kung mediated sa pamamagitan ng kanilang residential internet router, na gumagamit ng isang pamamaraan na tinatawag na network address translation (NAT). Ang suporta para sa dumarating na trapiko sa pamamagitan ng iba-ibang routers, at ibat ibang mga diskarte ay binuo upang gumana sa ibat ibang uri ng mga router. Sa partikular, ang mga routers na nagpapatupad ng symmetric NAT ay hindi maaaring madaling magtrabaho sa paligid maliban kung malinaw na isinaayos upang suportahan ang iba pang mga mekanismo ng hole-punching tulad ng Internet Gateway Device Protocol (IGDP).

Ang Harmony's P2P layer ay sinusubukan upang makita ang mekanismo ng NAT sa likod kung saan ang isang node ay nagpapatakbo at gumagamit ng tamang mekanismo ng nakapaloob na trabaho, tulad ng STUN, TURN, IGDP, atbp. Sa partikular, ang Harmony ay nagpapatupad ng pangkalahatang detection and mitigation protocol na pinangalanang ICE (Interactive Connectivity Establishment).

6.5 Support for Locator Mobility

Maaaring palitan ng mga node ang kanilang mga IP address, na may ilang uri ng mga node nang higit pa kaysa sa iba. Ang isang halimbawa ay isang laptop, na maaaring madalas na lumukso sa pagitan ng ibat ibang mga network ng Wi-Fi, na may pagbabago ng IP address sa bawat oras. Kapag ang isang IP address ng isang node ay nagbabago, ang lahat ng umiiral na mga koneksyon sa transportasyon na gumagamit ng IP address bilang isang lokal o remote endpoint ay naaantala, at ang mga aplikasyon na direktang gumagamit ng naturang mga koneksyon sa transportasyon ay kailangang uling magtatag ng mga koneksyon gamit ang bagong IP address upang magpatuloy. Ang gayong isang paghahatid ng koneksyon ay mahirap ipatupad nang wasto ng may kaunting pagbaba ng serbisyo sa application-layer. Gayundin, ang paghawak ng paghahatid ng koneksyon ay kadalasang nakakapinsala sa mga protocol ng application-layer (tulad ng mga protocol na pinagbasehan ng pinagkasunduan).

Ang network layer ng Harmony, upang malutas ang problemang ito, ay nagpapakilala ng isang malinis na paghihiwalay sa pagitan ng pagkakakilanlan ng node (cryptographic key pair na nagmamay-ari ng node) at tagahanap ng node (na tumatanggap ng network/ transport-layer kung saan maabot ang node) gamit ang industry-standard Host Identity Protocol Version 2 (HIPv2). Ang HIPv2 ay nagbibigay-daan sa mga tagahanap ng isang node na baguhin sa paglipas ng panahon habang pinapanatili ang pagkakakilanlan ng node, sa pamamagitan ng pagbibigay ng mga mekanismo para sa pagtuklas ng tagahanap, node-to-node na kaugnayan

sa seguridad, at tunneling ng trapiko sa itaas na layer na nauugnay sa local/ remote node identity bilang katapusan.

7. Incentive Model

7.1 Mga Gantimpala ng Pinagkasunduan

Matapos ang matagumpay na pangako ng isang block, ang isang protocol na tumutukoy sa bilang ng mga bagong mga token ay igagantimpala sa lahat ng mga validator na pumirma sa block na proporsyon sa kanilang voting shares. Ang mga bayarin sa transaksyon ay igagantimpala din tulad sa mga validator.

7.2 Stake Slashing

Para sa anumang mga hindi tamang pagkilos na nakita ng network, ang isang tiyak na halaga ng mga naka-stake na mga token ay i-slashed. Halimbawa, kung ang isang leader ay nabigo upang tapusin ang proseso ng pinagkasunduan at nag-alarma ng proseso ng pagbabago ng leader, ang P_{vote} staked na mga token ay i-slashed. Kung ang mga validator ay napatunayan na nag-sign ng isang hindi tapat na block, ang lahat ng kanilang stake sa ilalim ng parehong shard ay i-slashed. Ang matinding kaparusahan na ito ay sinadya upang mapigilan ang anumang hindi tapat na pag-uugali at gawing secure ang network hangga't maaari. Ang isang patunay ng maling paggawi ay maaaring maging dalawang signed blocks na nagkakasalungatan sa bawat isa. Ang alinmang validator ay maaaring magsumite ng isang transaksyon upang patunayan ang masamang asal ng iba pang validator at kung napatunayan na, ang slashed token ay igagantimpala sa mga tagapagpatunay.

7.3 Stake withdrawal

Long-range Attacks

Ang proof-of-stake blockchains, ay hindi tulad ng proof-of-work blockchains, na may posibilidad na magdusa mula sa mga pag-atake ng malayuan. Ang mga ito ay mga pag-atake na nakakuha ng katunayan na ang mga patunay ay batay sa mga lagda sa halip na sa mga mapagkukunan-masidhing gawain. Sa pag-atake ng malayuan ang mga private key ng tapat na mga validator ay ninakaw nang matagal pagkatapos na magamit na nila, at ang magsasalakay ay makagawa ng isang forked blockchain sa pamamagitan ng pagpirma ng pekeng mga block sa mga susi na iyon. Kapag nangyari ito, ang mga bagong validator na sumali sa network ay walang paraan upang makilala ang pagkakaiba sa pagitan ng orihinal, legitimate chain at simulation chain ng magsasalakay.

Ang mahabang hanay ng pag-atake ay nangyayari sa mga sumusunod na dalawang sitwasyon. Maaaring nakompromiso ang private key sa pamamagitan ng kakulangan ng seguridad sa mga validator, o mas karaniwan, sa pamamagitan ng katotohanan, pagkatapos ng isang validator na bawiin ang kanilang token, maaari siyang makinabang sa pananalapi kung ang isang magsasalakay na naghahanap upang bumili ng private key nito. Gayundin, sa pamamagitan ng pagdisenyo ng bawat hanay ng mga validator ay pinagkakatiwalaang aprubahan ang block ng mga transaksyon na tumutukoy din sa susunod na hanay ng mga validator. Matapos ang sapat na private key (i.e. mga sama-sama na humawak ng higit sa $\frac{2}{3}$ voting shares sa isang shard) ay nakompromiso, ang isang magsasalakay ay may ganap na kontrol sa kung sino ang kasunod na mga validator.

Long-range Defense: Resonant Quorums

Pinoprotektahan ng proof-of-work blockchain laban sa itaas na mga pag-atake sa pamamagitan ng pagbibigay ng tapat na mga validator ang isang layunin na paraan sa pagpili ng fork. Sa isang proof-of-work blockchain, ang pagpili ng fork upang piliin ang canonical chain ay ang naipon na halaga ng trabaho na ginawa sa mga tuntunin ng mga nacompute na hash.

Sa isang proof-of-stake blockchain, ang tanging layunin na panukalang maaaring magamit upang pumili sa pagitan ng mga fork ay ang kabuuang timbang ng mga lagda na ginamit upang aprubahan ang bawat block. Kung gagamitin natin ang mga weighted signature upang ihambing ang dalawang magkakaibang mga block, pumupunta kami sa sumusunod na equation upang matukoy kung kailan maaaring magfork ang isang chain:

Safety = “Block approval key weight” - “Compromised key weight”

Ang "Block approval key weight" ay nangangahulugan ng kapangyarihan ng pagboto ng mga keys na naka-sign sa block. Kung, sa pamamagitan ng stake weight, mas maraming private keys ang nakompromiso kaysa sa ginamit upang aprubahan ang isang block, kung gayon ang block ay maaaring magkahiwalay. Hanggang pagkatapos, ang mga validator ay laging gusto ang orihinal, lehitimong bersyon ng block.

Ang Harmony ay nagpapakinabang sa kaligtasan ng bawat block sa proof-of-stake blockchain sa pamamagitan ng pag-maximize ng equation na ito. Ito ay hindi maaaring gawin upang disincentivize pagkasira ng private keys sa mahabang panahon. Ang Harmony sa halip ay nagpapasigla ng mga validator upang ma-maximize ang approval weight ng bawat block matapos makamit ang isang quorom. Ginagawa ito sa pamamagitan ng pag-aatas sa mga validator na mag-sign sa bawat block na inaprobahan ng quorom bago pahintulutan ang mga validator na bawiin ang kanilang stake. Ang mga bagong karagdagang mga lagda ay kailangan lamang na umiiral sa loob ng blockchain, at hindi nila kailangan na mabuo sa oras ng pinagkasunduan para sa bawat block. Dahil dito, ang mga bagong lagda ay maaaring idagdag

sa susunod na mga block kapag ang mga validator ay nagpasiya na bawiin ang kanilang stake, at sa gayon maaari nilang malayang mapabuti ang kaligtasan ng chain na hindi nakakaapekto sa kursong ito.

8. Future Research

8.1 Fraud Proofs

Ang kakayahan ng pagpapatunay ng hindi magandang asal ng mga validator ay mahalaga para sa isang light client na pinagkakatiwalaan ang block data na kanilang natanggap. Sa kaso ng komunikasyon ng cross-shard, ang bawat shard ay isang light client ng iba pang mga shards. Ang pagtiyak na ang mga mensahe na ipinadala sa pagitan ng mga shards ay maaasahan na napakahalaga para sa pagkakapare-pareho ng datos sa shard. Kami ay aktibong nagsasaliksik sa paksa na maaaring pagkunan ng mga datos [29] at mga pruweba sa pandaraya [2] upang patibayin ang seguridad ng aming protocol.

8.2 Stateless Validators

Sa isang mataas na throughput ng blockchain, ang laki ng datos ng blockchain ay lalago ng mas mabilis kaysa sa umiiral na mga chain, na isang malaking problema para sa mga bagong validator upang mabilis na i-sync. Ito ay nagpapahirap sa proseso ng resharding dahil kung ang mga bagong validator ay hindi maaaring i-sync sa oras, pagkatapos ay ang quorom ng mga validator na maaaring hindi maabot para sa isang bagong block upang maaprubahan, at kahit na ang quorom ay matugunan, ang seguridad ng protocol ay maaaring nabawasan. Ang state block pruning ay isang nagpapagaan sa problema, ngunit hindi ito sulit dahil ang estado mismo ay maaaring lumaki ng mas malaki. Kami ay aktibong naghahanap sa magpapagana ng stateless client [30,31] kung saan ang mga validator ay hindi kailangang i-sync ang buong estado upang mapatunayan ang mga transaksyon.

Mga Sanggunian

[1] J.R. Douceur, The Sybil attack, in: 1st International Workshop on Peer-to-Peer Systems (IPTPS 02), 2002.

[2] Al-Bassam, M., Sonnino, A., & Buterin, V. (2018). Fraud Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities. CoRR, abs/1809.09044.

[3] Vasin, P. (2014) Blackcoin's Proof-of-Stake Protocol v2, https://blackcoin.co/blackcoin-pos-protocolv2-whitepaper.pdf

[4] A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. Cryptology ePrint Archive, Report 2016/889, 2016. http://eprint.iacr.org/.

[5] P. Daian, R. Pass and E. Shi, Snow White: Robustly reconfigurable consensus and applications to provably secure proofs of stake, Cryptology ePrint Archive, Report 2016/919, 2017. [6] Rafael Pass and Elaine Shi. Thunderella: Blockchains with optimistic instant confirmation. https://eprint.iacr.org/2017/913.pdf.

[7] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: A Fast Blockchain Protocol via Full Sharding." Cryptology ePrint Archive, Report 2018/460, 2018. https://eprint.iacr.org/2018/46 0. [8] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in 2018 IEEE Symposium on Security and Privacy (SP), pp. 19–34, 2018.

[9] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pages 17–30, New York, NY, USA, 2016. ACM.

[10] George Danezis and Sarah Meiklejohn. Centrally banked cryptocurrencies. In 23rd Annual Network and Distributed System Security Symposium, NDSS, 2016.

[11] The QuarkChain Team. Cross Shard Transaction. https://github.com/QuarkChain/pyquarkchain/wiki/Cross-Shard-Transaction

[12] The Zilliqa Team. The zilliqa technical whitepaper. https://docs.zilliqa.com/whitepaper.pdf, August 2017.

[13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. Available at https://bitcoin.org/ bitcoin.pdf.

[14] Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI '99), New Orleans, Louisiana, February 1999.

[15] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In Proceedings of the 25th USENIX Conference on Security Symposium, 2016.

[16] Drijvers, M., Edalatnejad, K., Ford, B., & Neven, G. (2018). Okamoto Beats Schnorr: On the Provable Security of Multi-Signatures. IACR Cryptology ePrint Archive, 2018, 417.

[17] B. Alangot M. Suresh A. S Raj R. K Pathinarupothi K. Achuthan "Reliable collective cosigning to scale blockchain with strong consistency" Proceedings of the Network and Distributed System Security Symposium (DISS'18) 2018.

[18] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. Cryptology ePrint Archive, Report 2017/454, 2017.

[19] Baruch Awerbuch and Christian Scheideler. Towards a scalable and robust DHT. In Proceedings of the Eighteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '06, pages 318–327, New York, NY, USA, 2006. ACM.

[20] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In CRYPTO 2018, 2018.

[21] D. Boneh, B. B¨unz, and B. Fisch. A survey of two verifiable delay functions. Cryptology ePrint
Archive, Report 2018/712, 2018. https://eprint.iacr.org/2018/71 2.

[22] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science, SFCS '87, pages 427–438,
Washington, DC, USA, 1987. IEEE Computer Society.

[23] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In 37th IEEE Symposium on Security and Privacy, May 2016.

[24] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. CoRR, abs/1710.09437, 2017.

[25] E. Syta, P. Jovanovic, E. Kokoris-Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford. Scalable Bias-Resistant Distributed Randomness. In 38th IEEE Symposium on Security and Privacy, May 2017.

[26] T. Hanke, M. Movahedi, and D. Williams. Dfinity technology overview series consensus system, January 2018.

[27] The Ethereum Foundation. Ethereum Whitepaper. https://github.com/ethereum/wiki/wiki/White-Paper.

[28] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information
Security: Advances in Cryptology, ASIACRYPT '01, pages 514–532, London, UK, UK, 2001. Springer-Verlag. https://www.iacr.org/archive/asiacrypt2001/22480516.pdf

[29] The Ethereum Team. A note on data availability and erasure coding.
https://github.com/ethereum/research/wiki/A-note-on-data-availability-and-erasure-coding

[30] A. Chepurnoy, C. Papamanthou, Y. Zhang. Edrax: A Cryptocurrency with Stateless Transaction Validation. Cryptology ePrint Archive, Report 2018/968.

[31] V. Buterin. The Stateless Client Concept. https://ethresear.ch/t/the-stateless-client-concept/172

[32] Derek Leung, Adam Suhl, Yossi Gilad, and Nickolai Zeldovich. Vault: Fast bootstrapping for cryptocurrencies. Cryptology ePrint Archive, Report 2018/269, 2018.

[33] Ethereum Wiki. On Sharding Blockchain.

https://github.com/ethereum/wiki/wiki/Sharding-FAQs [34] M. F. Nowlan, J. Faleiro, and B. Ford. Crux: Locality-preserving distributed systems. CoRR, abs/1405.0637, 2014.

[35] George Danezis and Sarah Meiklejohn. Centrally banked cryptocurrencies. In 23nd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016. The Internet Society, 2016.

[36] Paul Dworzanski. A note on committee random number generation, commit-reveal, and last-revealer attacks. http://paul.oemm.org/commit_reveal_subcommittees.pdf.

[37] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In Revised Papers from the First International Workshop on Peer-to-Peer

Systems, IPTPS '01, pages 53–65, London, UK, UK, 2002. Springer-Verlag.

[38] David K. Gifford, Doctoral Dissertation, Information Storage in a Decentralized Computer System.

[39] Prince Mahajan, Lorenzo Alvisi, and Mike Dahlin Consistency, Availability, and Convergence ,

Technical Report (UTCS TR-11-22) http://www.cs.cornell.edu/lorenzo/papers/cac-tr.pdf

[40] Wyatt Lloyd et. al, Don't Settle for Eventual: Scalable Causal Consistency for Wide-Area Storage with COPS, Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP'11). https://www.cs.cmu.edu/~dga/papers/cops-sosp2011.pdf

[41] Peter Bailis†, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica, Bolt-on Causal Consistency, [SIGMOD'13].

[42] Postel, J. (1981). Transmission control protocol specification. *RFC 793*.

[43] Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., & Minder, L. (2011). RaptorQ forward error correction scheme for object delivery (No. RFC 6330).