Date	Start Time (24h)	End Time (24h)	Duration (hours)	Week Subtotal	Week Overtime	Notes
2021-07-17	1:00	2:00	1			Token and NFT experimentation and debugging; Client API implentation for token operation
2021-07-17	18:00	0:00	6			Research on daVinci market; Client: ERC20 grid; Token assets manager; Unified token key compute utility; In-browser IPFS retrieval analysis and benchmark; Client API initialization for token contracts and metadata, and API implementations for metadata retrieval and balance checking for token and NFT contracts; Refactor and modularization of client wallet components; Smart contract support for Override-Track capability; Smart contract optimization for universal deployability (<24K); Smart contract multi-track capability; Client side state management (reducer, saga, actions) for tracking tokens, token balances, and currently selected tokens; Direct support for well known ERC-20 tokens; Infrastructure support for tokens with custom decimal settings; Unifty bn.js dependecy; Multi-network support for token operations and views; Token operation hash and commit hash utilities, and debugging; Auto chaining and auto refresh token balance after commit-reveal flow; Support sending tokens; fix bugs related to reveal token operations; Distinguished UI for sending tokens (re-used from sending ONEs); Various bug fixes; Complete flow demo + verification;
2021-07-18	0:00	5:00	5			(Continued)
2021-07-18	16:00	0:00	8			NFT Grid, support for both ERC721 and ERC1155, share same underlying token abstraction with ERC20; Support image, animation, and metadata rendering; NFT details toggle; Responsive grid; End-to-end debugging and testing for tokens; Upgrade versions and warning messages for older clients; v0.3 release; Separate view and filtering for ERC20 and NFT grids; Support sending NFT through existing transfer UI;
2021-07-19	0:00	3:00	3			(Continued)
2021-07-20	17:00	18:00	1			OTP Input and Refocus code review + revision + bug fixes; Provide feedback; Fix missing name issue for some ERC721 tokens
2021-07-23	18:00	0:00	6	30		Address critical security vulnerabilities with commit-reveal (#47); Constant-time commit lookup (#3, #4); Implement suggested contract optimizations in Common Prefix preliminary report (immutable variables, unchecked arithmetics); Added NatSpec compliant comments in contract code; Fix an issue with incorrect key computation of tracked tokens; Unify all reveal operations into a single function with different operation types;
Week 13	**7/24**	**7/30**				
2021-07-24	18:00	0:00	6			(Continued) Core library upgrade and new commit hash computation utlities; Relayer support for new contract and unified reveal; Refactored relayer routes and redirects; Dual version support and backward compatibility of relayer; Wallet specific client header in client-relayer communication; Fix issues with Truffle providers for local debugging and testing; Fix all local core library and smart contract tests
2021-07-25	0:00	5:00	5			(Continued) SecureFlow core library for new commit-reveal mechanism, plus backward compatibility; Modulelized relayer; Fix bugs with previous versions of commit-reveal flow
2021-07-25	15:30	16:30	1			Review of new DoS security issue with commit-reveal mechanism
2021-07-25	15:00	0:00	9			Analysis and response to Common Prefix security audit preliminary report, end-to-end; Revisit Ivan's proposal for commit-reveal (#4) and analyze vulnerability for man-in-the-middle attacks; Analysis and response to feedback on token tracking mechanism (comment, #57); Reply and analysis of Ivan's reviews on Client Security (#58)
2021-07-26	4:00	6:00	2			(Continued) and creating a solution to DoS vulnerability of the new commit-reveal mechanism (#59)
2021-07-26	6:00	9:30	3.5			(Continued) Completing the analysis and response for preliminary report
2021-07-26	19:00	20:00	1			Review of contract breakdown (Ivan, #57). General repository maintainence and refactor; Preliminary implementation of better commit-reveal mechanism that addresses DoS vulnerability #59; Further analysis and response to token tracking using smart contract v.s. using API (#57)
2021-07-26	20:00	22:30	2.5	30		Implementation of anti-DoS, privacy preserving commit-reveal mechanism with parameter hash and verification hash
2021-07-26	22:30	0:00	1.5			(Continued)
2021-07-27	0:00	4:30	4.5			(Continued) Tests for new commit mechanisms; Core library for computing verification hash; Upgrade contracts to v7; Relayer implementation for v7 contracts and backward compatibility; Debugging and adding deprecated function selectors to contracts for backward compatibility; Client support for v7 contracts and backward compatibility; Deploy v0.4.1
2021-07-27	15:00	18:00	3			More review of Ivan's contract simplification proposal (#2); More analysis and reply to Ivan's review on Client Security design (#58); Detailed analysis and revision of scrambled memory layout methods
2021-07-27	18:00	20:00	2			Staking pool design and proposal (#62); Analysis of daVinci marketplace design and smart contracts and identify gaps with EIP-1271 (#61)
2021-07-30	10:00	11:30	1.5			Sync with Shashank; Walk through with client security methods; Identify issues with scrambled memory layout method; Discussions around potential use of verifiable delay functions
2021-07-30	21:30	0:00	2.5		15	Sync with Ivan and discuss issues with scrambled memory layout methods; Revision of scrambled memory layout method for client security (#63)

Week 14	7/31	8/6				
2021-07-31	0:00	1:00	1			Token tracking approach explanation, analysis, and comparisons with traditional methods and their problems
2021-07-31	11:00	16:00	5			Review, research, analysis, and reply on Common Prefix report draft and analysis on client security
2021-07-31	22:00	0:00	2			Argon2 webassembly implementation research, testing, analysis, and implementation for speed optimization and batch processing (github: polymorpher/argon2-browser)
2021-07-31	0:00	2:00	2			(Continued)
2021-08-01	18:00	20:00	2			Argon2 webassembly debugging, benchmarking, bug fixes (hash buffer length mismatch), progress observer and npm-compatible structure and version publishing for customizations
2021-08-01	21:00	0:00	3			Implementation of client security: controlled randomness, double otp, and initial integration with customized argon2
2021-08-02	0:00	5:00	5			(Continued) (#64) core library implementation of client security: merkle tree computation, recover randomness, worker messages, sha256 batch interface, and upgrade related functions to async; Implementation of using AES counter mode for seeded random generation; Bitshifting for controlled randomness; Tests for client security related functionalities by themselves and with smart contract
2021-08-02	14:00	18:00	4			Detailed analysis, research, and reply on Ivan's reviews on client security attack models (#58) - comprehensive breakdown into different levels of read/write/temporal access in real world scenarios and device limitations; reply and review of suggestions to smart contract (#60); reply to other comments (nonce, #58) and discussions on Telegram
2021-08-03	17:00	18:00	1			Review double OTP frontend UI implementation from Haolin (#65)
2021-08-03	23:00	0:00	1			(SlowMist audit) Reentrance vulnerability analysis and bug fixes; (Hackathon) Developer environment guide; README and setup guide for all components (repostiroy wide, smart contract, relayer, web client)
2021-08-04	0:00	4:00	4	30		(Continued) repository refactoring, fixing patch scripts, command examples; Review and finalize feedback on Haolin's work on double OTP frontend UI (#65)
2021-08-04	22:00	0:00	2			Complete remaining end-to-end implementation for double OTP and client security methods; end-to-end tests, benchmarking, bug fixes of client security; Automated tests with smart contract integrations for all implemented client security methods
2021-08-05	0:00	9:00	9			Core library: incorporate client security methods and parameters; Allow arbitrary paramsHash (through randomized data field) in recover operation to preserve privacy; Deprecate brute-force recovery method; Fix bugs with compute recovery hash; New recovery flow implementation; Update all smart contract tests; Relayer: better debugging flow and tools; Client security parameter tuning and default set; Enable client-security by default in client (#66); Complete frontend UI integration with new core library; Deploy to production and end-to-end tests; Address minor issues and suggestions in SlowMist audit report; Correct SlowMist audit report error; Review and finalize SlowMist audit report; Further discussions and analysis with Ivan on the potential use of Trusted Execution Environment and general security expectations in cases when attacker gains access to OTP just-in-time
2021-08-05	21:00	0:00	3			Version-based client security parameters; Sync versions in all components; Worker-based randomness recovery implementation to enable undisrupted user experiencei in performing transactions with client security enabled, such that heavy computation in randomness recovery becomes unnoticable to users; Simply all warning messages; Fix bugs with incorrect warning messages due to unsynced wallet state between client and blockchain
2021-08-06	0:00	2:30	2.5			(Continued) and, research and analysis on Argon2 speed issues, necessity, and areas of improvement (#67)
2021-08-06	3:00	5:00	2		18.5	Further review and detailed reply to Common Prefix audit report on unresolved errors and issues in the audit report
Week 15	**8/7**	**8/13**				
2021-08-07	0:00	4:00	4			argon2 C code optimization, profiling, and debugging; build options and compiler optimizations; experimentation of SIMD; interfacing with web assembly and javascript sides (github:polymorpher/argon2#batch); End-to-end testing in-browser using different runtime environments (node beta, node standard, safari beta, safari standard, Chromium); Notes and findings (#16); In-depth research into WebAssembly cause of slowness, pitfalls, and general expectations (memory layout, allocation method, instruction sets, compilation flags)
2021-08-07	13:00	18:00	5			(Continued), and backward compatible support for legacy recovery methods to support older wallets
2021-08-08	0:00	1:00	1			Call for developers; Further developer documentation
2021-08-08	11:00	15:00	4			Detailed analysis and reply on further issues and errors in Common Prefix audit report
2021-08-09	11:00	12:30	1.5			More research and discussions on Common Prefix audit report - miner rentability, GPU hash power global proportion, ASIC compatibility for brute-force 1wallet, cost analysis method and practicality; Minor errors and suggestions for earlier sections

2021-08-09	2:30	3:30	1			(#70) Review, research, and analysis on proof of stake safety guarantee for 1wallet operations at large scale, economic incentives, potential pitfalls, and practical gas cost for common operations
2021-08-10	1:30	2:30	1			Binance withdrawal issues investigation and report (#71)
2021-08-10	14:00	15:00	1			Emergency patch for loss of access to newly created single OTP wallets (#72); Warning messages for buggy wallet version; Attempt to fix and debugging Binance withdrawal issues through tuning contract payment receiving function
2021-08-10	19:00	20:00	1			Fix issues with some ERC20 tokens' balance being incorrectly displayed due to custom decimal settings
2021-08-12	0:00	5:00	5			Research and planning for upcoming changes: dApp integration methods (#73), pros and cons of methods used in other wallets (#73); Documentations and tutorials required (#74); Upcoming changes for frontend and UI (#75), domain names (#76)
2021-08-12	14:00	17:00	3			Research and planning on changes to smart contract, upgradability, auto-recovery, and security implications (#78)
2021-08-13	0:00	2:00	2			Smart contract cleanup and minor logic improvements (#60, #68). Review and feedback on frontend UI changes on wallet name and address components (#77)
2021-08-13	21:00	21:30	0.5	30		Further review on frontend, completing the changes required, and apply address components everywhere (#77)
			Regular	120		
			Overtime		33.5	For meeting deadline set for security audit (end of July), shortened deadlines for client security, and urgent, necessary preparations needed for the upcoming Hackathon
			Total	153.5		