Tabriz Journal of Electrical Engineering (TJEE), vol. 53, no. 1, Spring 2023 Serial no. 103
DOI: 10.22034/tjee.2023.15981

A Protocol for Authenticated Anonymous Communications by Post-Quantum Cryptography and Smart Contracts

Mohammad Mahdi Mojahed[1], MSc; Amir Hassani Karbasi[2], PhD; Sadegh Dorri Nogoorani[3*], Assistant Professor; Alireza Kiakojouri[4], MSc

1- Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran, E-mail: m.mojahed@modares.ac.ir.
2- ISMS Lead Auditor and Industrial Cybersecurity Consultant, Amnafzar Gostar Apadana Co., Tehran, Iran, E-mail: karbasi@amnafzar.net.
3- Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran, E-mail: dorri@modares.ac.ir.
4- Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, E-mail: a.kiakojouri@aut.ac.ir.

Short Abstract
Security of communications is a foundation for interactions in the cyber space. Recent advances in the field of quantum computing has attracted attentions to quantum attacks. Post-quantum cryptography is a relatively new field of research and few post-quantum protocols have been proposed for secure communications. In particular, authentication of the two communicating peers while preserving their privacy and anonymity is a real challenge. In this paper, we propose a comprehensive protocol for secure authentication, key agreement, and message encryption which is resistant to quantum attacks. We use the blockchain technology and a smart contract for authentication, and the double-ratchet protocol for end-to-end encryption. Our initial key agreement uses post-quantum cryptography which brings a high level of security to our protocol. We store public keys on a cloud storage for saving costs but authenticate them using smart contracts. Our analysis of the proposed protocol demonstrates our superiority from privacy, security, and performance aspects in comparison to the related works.

Keywords
Post-quantum cryptography, blockchain, smart contract, privacy, anonymity, key management.

1- Short Introduction

With the advent of quantum computers and the threats posed to current cryptographic algorithms, this paper attempts to use two emerging technologies, blockchain and post-quantum algorithms, to provide a secure end-to-end communication. In establishing a connection between two anonymous entities, always identifying the two entities while preserving their privacy is the most challenging part of the communication. Anonymity means the possibility of authenticating the parties to each other while being anonymity from the point of view of the others. During the communication, some keys are exchanged between the parties for secure communication, and the management of these keys is also important.

2- Proposed Work and Methodology

In our protocol, a quantum-resistant double-ratchet cryptographic protocol is used for secure messaging, and an Ethereum smart contract for authentication. In particular, each user has two public key pairs; a post-quantum (SIDH) key pair for the initial handshake, and a classic (ECDH) key pair for successive key generations. We take a hybrid approach [A] in which both keys are involved in the protocol, so that the SIDH algoritm protects against quantum attacks, while the ECDH key brings about efficiency, and proven classic security. The SIDH public key is retrieved from a cloud storage (IPFS), and verified by a server and an Ethereum smart contract which completes the authentication. In this protocol, anonymity and privacy of all entities are preserved throughout the authentication process and the result of the authentication is reliable.

Our protocol improves two existing works [C, D]. In [C], the idea of integrating post-quantum cryptography into the double-ratchet protocol [B] has been proposed, and the use of a server and a blockchain smart contract for authentication is proposed in [D]. In this paper, the server is less involved in the processes and saves less information. Our storage space in the blockchain, our gas consumption, and the number of transactions required to perform the tasks by our smart contract have been reduced, and some security vulnerabilities in the previous smart contract [D] are fixed. Reducing the number of transactions has seriously reduced gas consumption and also simplified protocol processes. Also, modifying the key storage process has made the protocol more practical so that this protocol can be used in real operation.

We integrated the Microsoft implementation of SIDH (in C) into our Python implementation of the double-ratchet and X3DH protocols. Our smart contract is also implemented in Solidity (the Ethereum network). Our evaluations demonstrate that while the overhead of the post-quantum part is much greater than the classic counter-part, but it can be completed in real-time (343ms), and the relative overhead reduces as the number of exchanged messages increase. Our gas consumption in the smart contract for a key owner and the server is also reduced by 93% and 48%, respectively, in comparison to [D], but the introcution of inquiry fee has increased our gas consumption of this transaction by 46%.

3- Conclusion

We introduced a complete quantum-resistant protocol for secure messaging which uses a blockchain smart contract for the authenticity of public keys. The protcol preserves the privacy of the involved parties and is secure against classic and quantum attacks. We also implemented our protocol and demonstrated its practicality in real-time applications.

4- References

A. N. Bindel, U. Herath, M. McKague, and D. Stebila, "Transitioning to a quantum-resistant public key infrastructure," in International Workshop on Post-Quantum Cryptography, Springer, 2017, pp. 384-405.
B. J. Alwen, S. Coretti, and Y. Dodis, "The double ratchet: Security notions, proofs, and modularization for the signal protocol," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2019, pp. 129-158.
C. A. H. Karbasi and S. Shahpasand, "SINGLETON: A lightweight and secure end-to-end encryption protocol for the sensor networks in the Internet of Things based on cryptographic ratchets," The Journal of Supercomputing, pp. 1-39, 2020.
D. H. Karbasi and S. Shahpasand, "A post-quantum end-to-end encryption over smart contract-based blockchain for defeating man-in-the-middle and interception attacks," Peer-to-Peer Networking and Applications, pp. 1-19, 2020.

پروتکلی برای ارتباطات گمنام احراز اصالت شده با رمزنگاری پساکوانتومی و قراردادهای هوشمند

محمد مهدی مجاهد[۱]، کارشناس ارشد؛ امیر حسنی کرباسی[۲]، دکتر؛ صادق دری نوگورانی[۳*]، استادیار؛ علیرضا کیاکجوری[۴]، کارشناس ارشد

۱- دانشکده مهندسی برق و کامپیوتر- دانشگاه تربیت مدرس- تهران- ایران- m.mojahed@modares.ac.ir
۲- شرکت امن افزار گستر آپادانا- تهران- ایران- karbasi@amnafzar.net
۳- دانشکده مهندسی برق و کامپیوتر- دانشگاه تربیت مدرس- تهران- ایران- dorri@modares.ac.ir
۴- دانشکده ریاضی و علوم کامپیوتر- دانشگاه صنعتی امیرکبیر- تهران- ایران- a.kiakojouri@aut.ac.ir

چکیده

ارتباط امن یکی از پایه‌ای‌ترین زیرساخت‌ها در فضای مجازی است. هرچند پروتکل‌های قدرتمندی به این منظور ارائه شده‌اند اما با پیشرفت‌هایی که خصوصاً در چند سال اخیر در ایجاد رایانه‌های کوانتومی اتفاق افتاده، مقاومت نسبت به حملات کوانتومی مورد توجه ویژه قرار گرفته است. به دلیل جدید و نابالغ بودن حوزه پساکوانتوم، پروتکل‌های پساکوانتومی معدودی در دسترس هستند. همچنین احراز هویت طرفین در ضمن حفظ حریم خصوصی و گمنامی آن‌ها همواره با چالش‌هایی همراه بوده است. در این مقاله، یک پروتکل کامل برای احراز هویت، توافق کلید، و تبادل پیام ارائه شده است که نسبت به حملات کوانتومی مقاوم است، از زنجیره قالب‌ها و قراردادهای هوشمند برای احراز هویت استفاده می‌کند، و با استفاده از الگوریتم چرخ‌دنده دوتایی و رمزنگاری انتها-به-انتها از امنیت بالایی در تبادل پیام برخوردار است. در این پروتکل از توافق کلید پساکوانتومی در آغاز ارتباط استفاده می‌شود. کلیدهای عمومی مربوطه در فضای ابری نگهداری، و برای احراز اصالت آن‌ها از قرارداد هوشمند استفاده می‌شود. کلیدهای رمزنگاری با استفاده از پروتکل چرخ دنده دوتایی تولید می‌شوند و برای هر پیام یکتا هستند. ارزیابی این پروتکل نشان می‌دهد نسبت به پروتکل‌های پیشین بهبود یافته، و توانسته است در عین حفظ حریم خصوصی و امنیت بالا، کارایی خوبی داشته و در عمل قابل استفاده باشد.

کلمات کلیدی

رمزنگاری پساکوانتومی، زنجیره‌ی قالب‌ها، قرارداد هوشمند، حفظ حریم خصوصی، گمنامی، مدیریت کلید.

نام نویسنده مسئول: دکتر صادق دری نوگورانی
ایمیل نویسنده مسئول: dorri@modares.ac.ir

تاریخ ارسال مقاله: ۱۴۰۱/۰۱/۲۵
تاریخ(های) اصلاح مقاله: ۱۴۰۱/۰۴/۲۵
تاریخ پذیرش مقاله: ۱۴۰۱/۱۱/۱۱

۱- مقدمه

در ایجاد ارتباط بین دو موجودیت گمنام همواره احراز هویت آن دو موجودیت با حفظ حریم خصوصی آن‌ها چالشی‌ترین بخش برقراری ارتباط است. در حین برقراری ارتباط کلیدهایی برای ارتباط امن بین طرفین رد و بدل می‌شود که مدیریت این کلیدها نیز اهمیت خود را دارد. منظور از احراز هویت گمنام، امکان احراز هویت طرفین ارتباط برای یکدیگر در عین حفظ گمنامی از دیدگاه سایرین است. پیش از این، پروتکل‌های مختلفی جهت انجام این امور معرفی شده‌اند. از آنجا که بسیاری از اطلاعات شخصی و شرکتی از طریق شبکه‌های اجتماعی و فضای وب ذخیره‌سازی و دست به دست می‌شود، نقض امنیت و دستیابی به محتوای این اطلاعات به صورت غیر رمز شده جذابیت زیادی برای مهاجمین دارد [۱].

فناوری زنجیره‌ی قالب‌ها، به علت حفظ و تأمین اصالت اطلاعات، گزینه مناسبی برای انجام این وارسی فراهم آورده است. به طور خاص می‌توان از قراردادهای هوشمند به این منظور استفاده کرد. قراردادهای هوشمند خودکار هستند و کد آن‌ها غیر قابل تغییر است. اما با توجه به شفافیت و غیرقابل انکار بودن جزئیات تراکنش‌ها و وضعیت قراردادهای هوشمند روی زنجیره‌ی قالب‌ها، عملیات احراز هویت باید ضمن امنیت، ضامن گمنامی نیز باشد.

به علت توجه شرکت‌های بزرگ و صنعتی و قطب‌های علمی جهان به حوزه‌ی رایانش کوانتومی انتظار می‌رود در آینده شاهد رایانه‌های کوانتومی با قدرت بسیار بالا باشیم. هرچند افزایش طول کلید الگوریتم‌های رمزنگاری قالبی [۲] می‌تواند امنیت نسبی برای این الگوریتم‌ها نسبت به حملات کوانتومی فراهم کند، اما این رایانه‌ها تهدید بزرگی برای رمزنگاری کلید عمومی متداول هستند و موجب به خطر افتادن امنیت پروتکل‌های ارتباطی امروزی می‌شوند. برای مقابله با تهدیدات رایانه‌های کوانتومی نیاز به یک پروتکل پساکوانتومی است که در رایانه‌های امروزی و در کنار الگوریتم‌های رمزنگاری کلاسیک قابل استفاده باشد تا بتوان به‌راحتی و به‌صورت گسترده و کارا از آن استفاده کرد. همچنین با توجه به نوظهور بودن الگوریتم‌های پساکوانتومی پیشنهادی و عدم ارائه استاندارد برای این نوع الگوریتم‌ها باید پروتکل پیشنهادی امکان مقاومت در برابر حملات کلاسیک را نیز داشته باشد و امنیت آن اثبات‌پذیر باشد. از طرفی احراز اصالت موجودیت‌ها به‌صورت گمنام در پروتکل‌های رمزنگاری فعلی مشکلی است که می‌توان با استفاده از فناوری زنجیره قالب و قراردادهای

هوشمند به‌صورت بهینه آن را حل کرد. در مقاله حاضر، همه ملزومات برقراری ارتباط گمنامِ احراز اصالت شده، که در عین سرعت و کارایی نسبت به حملات کوانتومی هم مقاوم باشد، در کنار هم دیده و به طور عملی پیاده‌سازی شده است. به این منظور:

- برای آغاز ارتباط از توافق کلید پساکوانتومی استفاده می‌شود.
- برای احراز هویت طرفین از زوج کلیدهای نامتقارن پساکوانتومی، که کلید عمومی آن در فضای ابری (IPFS) ذخیره شده است، استفاده می‌شود.
- از قرارداد هوشمند و زنجیره‌ی قالب‌ها برای اطمینان از عدم دستکاری کلید عمومی استفاده می‌شود.
- از پروتکل چرخ‌دنده دوتایی[1] برای ایجاد و تغییر سریع کلیدهای متقارن جهت رمزگذاری محتوای ارتباط استفاده می‌شود.

بنابراین نوآوری‌های این مقاله عبارتند از:

۱) پروتکل پساکوانتومی کاملی که در برابر حملات کوانتومی مقاوم است و در عین حال کارایی مناسب جهت استفاده در ارتباطات متنی، صوتی و تصویری به‌صورت کارا را دارد.

۲) احراز اصالت موجودیت‌ها در عین حفظ گمنامی با استفاده از زنجیره‌ی قالب‌ها انجام می‌شود که امکان دستکاری، جعل، و حمله مرد میانی را برطرف می‌کند.

۳) الگوریتم‌های رمزنگاری استفاده شده به صورت پیمانه‌ای در پروتکل پیشنهادی قرار گرفته‌اند و امکان انتخاب بین آن‌ها وجود دارد (این موضوع در ارزیابی‌ها بررسی شده است.)

شایان ذکر است که پروتکل پیشنهادی به طور کامل برای ارزیابی دقیق‌تر پیاده‌سازی شده است و نتایج ارزیابی به صورت کمی و کیفی گزارش شده است.

در ادامه، ابتدا کارهای مرتبط و مفاهیم پایه را بررسی می‌کنیم. سپس در بخش ۲ به مفاهیم پایه می‌پردازیم. در بخش ۳ پروتکل پیشنهادی را معرفی می‌کنیم. در بخش ۴ به توضیح پیاده‌سازی و ارزیابی کارایی می‌پردازیم و نهایتا در بخش ۵ مقاله را جمع‌بندی می‌کنیم.

۱-۱- کارهای مرتبط

مؤسسه استاندارد آمریکا (NIST) یک مسابقه پنج ساله را در جهت تعیین یک استاندارد برای الگوریتم‌های رمزنگاری پساکوانتومی برگزار می‌کند. این فرایند استانداردسازی (مسابقه) از سال ۲۰۱۶ میلادی شروع شده است و مهلت ارسال اولیه طرح‌های پیشنهادی مربوط به پایان سال ۲۰۱۷ میلادی بوده است. این فرایند هنوز به پایان نرسیده است و در آخرین اعلامیه آن (۵ جولای ۲۰۲۲) چهار کاندیدا برای استانداردسازی معرفی شده‌اند و چهار کاندیدا به دور چهارم وارد شده‌اند. الگوریتم‌ها و پروتکل‌های پساکوانتومی مناسبی که قابلیت پیاده‌سازی داشته باشند در این رقابت معرفی شده‌اند [۵-۳]. در دور اول این فرآیند ۶۹ پروتکل مختلف مورد بررسی قرار گرفته‌اند که به گروه های Code-Based, Multivariate-Based, Lattice-Based, Hash-Based و Isogeny-Based تقسیم می‌شوند [۶].

Bindel و همکاران در [۷] در مورد مهاجرت از الگوریتم‌ها و پروتکل‌ها غیرکوانتومی به جایگزین پساکوانتومی آن‌ها تحقیقاتی انجام داده‌اند و راهکاری برای این مهاجرت و همچنین چالش‌های آن را مطرح کرده‌اند. یکی از چالش های مهم برای پساکوانتومی کردن پروتکل های رمزنگاری موجود، بروزرسانی حجم عظیمی از برنامه‌هایی است که از پروتکل‌های پراستفاده‌ای مانند زیر ساخت کلید عمومی (PKI[2]) استفاده می‌کنند و سازگاری برنامه‌ها با نسخه‌های جدید این پروتکل‌ها بایستی مدنظر قرار بگیرد. یکی از روش‌های ارائه شده در این مقاله استفاده از الگوریتم‌های ترکیبی-چندگانه است که هم بتوان امنیت برنامه ها را در برابر حملات کوانتومی و غیرکوانتومی تا حدی تضمین کرد و هم کارکرد این برنامه ها مختل نشود؛ چرا که از ایمنی پروتکل های جدید پساکوانتومی اطمینان کامل نداریم و از طرفی امنیت پروتکل‌های قدیمی نیز در برابر حملات کوانتونی مورد ابهام است. لذا از یک یا چند پروتکل پساکوانتونی به همراه یک پروتکل غیرکوانتومی به صورت ترکیبی استفاده می شود که در صورت شکست یکی، همچنان امنیت برقرار بماند. در این مقاله همچنین طرح گواهی امضای ترکیبی ارائه شده و چندین روش برای ترکیب این گواهی‌های امضا مورد بررسی قرار گرفته است. همچنین نشان داده شده است که این امضا غیرقابل جعل و جداناپذیر است. این مقاله طرح امضای ترکیبی را روی سه بخش پراستفاده از زیرساخت کلید عمومی و امضای دیجیتالی یعنی گواهینامه های (X509، TLS) و پروتکل امضای ایمیل (S/MIME) با رویکرد پیاده‌سازی و عملیاتی بودن استفاده از پروتکل‌های جدید در برنامه‌های موجود مورد بررسی قرار داده است.

همچنین تلاش هایی در [۸] برای پساکوانتومی کردن گواهینامه های X509 صورت گرفته است و چالش های مختلف الگوریتم های امضای پساکوانتومی از جمله طول کلید و امضای طولانی که باعث سربار انتقال در شبکه، تاخیر و هدر رفت پهنای باند می‌شود مورد بررسی قرار گرفته است.

در مرجع [۹] آزمایشی توسط شرکت گوگل انجام گرفته است و از روشی مشابه برای ارزیابی و بررسی الگوریتم‌های پساکوانتومی در محیط عملیاتی بهره گرفته شده است. در این مورد، این شرکت در مرورگر کروم از الگوریتم پساکوانتومی تبادل کلید NewHope [۱۰] بر روی الگوریتم رمزنگاری خم بیضوی[3] استفاده کرده است. با این کار در صورتی که الگوریتم پساکوانتومی دارای آسیب‌پذیری باشد جلوی به خطر افتادن امنیت اطلاعات توسط الگوریتم‌های خم بیضوی گرفته می شود و در صورتی که این الگوریتم مقاوم بماند، جلوی حملات غیرفعال کوانتومی[4] آینده را می‌گیرد. این آزمایش بر روی برخی از دامنه های گوگل مانند https://play.google.com انجام گرفته است و هدف اصلی آن ارزیابی الگوریتم‌های پساکوانتومی از لحاظ امنیت و کارایی در دنیای واقعی و در برابر حجم زیادی از اطلاعات است. همچنین در [۱۱] تحقیقات در مورد ارزیابی و بهبود کارایی الگوریتم های مسائل حلقه یادگیری با خطاها صورت گرفته است که نشان می‌دهد استفاده از این الگوریتم‌ها کاملا عملی است.

پروتکل چرخ‌دنده دوتایی در [۱۲] مورد بررسی قرار گرفته و ویژگی‌های امنیتی یک پروتکل رمزنگاری پیام‌رسانی ارزیابی شده است. پروتکل چرخ‌دنده دوتایی، پروتکل رمزنگاری پیام‌رسانی امن است که توسط پیام‌رسان‌های مطرحی چون Signal، WhatsApp، Google Allo، Skype و Facebook private messaging [۱۸-۱۳] استفاده می‌شود که کاربران آن‌ها به چند میلیارد نفر می‌رسد. پروتکل پیام‌رسانی OTR[5] [۱۹] زمینه‌ساز ارائه پروتکل چرخ‌دنده دوتایی بوده است که در آن از ایده چرخ‌دنده‌های مبتنی بر دیفی‌هلمن برای ایجاد کلیدهای جدید برای رمزنگاری هر پیام بهره برده شده است و بخش اصلی رمزنگاری در چرخ‌دنده دوتایی به‌شمار می‌آید. به دلیل اهمیت این پروتکل، در بخش پیش زمینه‌ها به تشریح جزئیات آن می‌پردازیم.

در زمینه پروتکل‌های رمزنگاریِ پساکوانتومیِ پیام‌رسانی و زنجیره قالب‌های مقاوم در برابر رایانه‌های کوانتومی کارهای خیلی کم‌تری انجام شده است. در صورتی که این دو حوزه مخاطبین بسیاری دارند (چندین میلیارد) و از اهمیت بالایی برخوردار هستند. در مرجع [۲۰] رایانه‌های کوانتومی حال حاضر بررسی شده‌اند و پروتکل پساکوانتومی چرخ‌دنده دوتایی با استفاده از پروتکل‌های دسته Isogeny از لحاظ کارایی مورد بررسی قرار گرفته است.

پروتکل Singleton برای رمزگذاری انتها-به-انتها در ارتباطات اینترنت اشیا (IoT) در [۲۱] معرفی شده است. در این پروتکل از الگوریتم چرخ‌دنده دوتایی استفاده و نشان داده شده است که در عین امنیت چنان کارا است که در حسگرهای کوچک نیز قابل استفاده است. پیاده‌سازی این پروتکل با استفاده از دو الگوریتم تبادل کلید دیفی هلمن خم بیضوی (ECDH) و دیفی هلمن توسعه‌یافته سه‌گانه[۶] (X3DH) پساکوانتوم مبتنی بر پروتکل SIDH مورد بررسی قرار گرفته است.

یک روش برای رمزنگاری انتها به انتها در کاربرد پیام‌رسانی فوری[۷] در [۲۲] ارائه شده است که برای مقابله با حملات مردمیانی و رهگیری از قراردادهای هوشمند استفاده می‌کند. در این روش از فایل سیستم همتا به همتای IPFS[۸] برای نگهداری کلیدهای عمومی و از قرارداد هوشمند برای اطمینان از اصالت و عدم دستکاری آن‌ها استفاده می‌شود. کلید رمزنگاری ارتباطات با ring-PAKE ایجاد می‌شود که یک پروتکل تبادل کلید مبتنی بر گذرواژه و مشبکه-مبنا (پساکوانتومی) است. روش کار به این صورت است که کاربر جدید از طریق پیام‌رسان خود یک زوج کلید نامتقارن تولید می‌کند و مقدار درهم کلید عمومی را در یک قرارداد هوشمند بر روی زنجیره‌ی قالب‌ها قرار می‌دهد. سپس اصل کلید عمومی را به طور امنی به یک کارگزار می‌دهد و از آن می‌خواهد که صحت مقدار درهم را در قرارداد هوشمند تأیید کند. کارگزار کلید عمومی را در IPFS ذخیره می‌کند و پس از بررسی مقدار درهم ذکر شده در قرارداد، آن را تأیید می‌کند. کاربر دیگری که قصد ارسال پیام دارد، درخواست خود را به کارگزار می‌دهد. کارگزار آدرس کلید روی IPFS و قرارداد هوشمند را در اختیارش می‌گذارد. کاربر با مراجعه به قرارداد هوشمند می‌تواند از اصالت و عدم دستکاری کلید مطمئن شود. سپس از این کلید عمومی استفاده می‌کند و بخشی از یک گذرواژه را به صورت رمزشده به مخاطب خود می‌فرستد. بخش دیگر هم با استفاده از پیامک (SMS) به وی داده می‌شود. نهایتاً با استفاده از ring-PAKE کلید جلسه برای ارتباط امن انتها به انتها رمز شده ساخته می‌شود.

۲- مفاهیم پایه

در این بخش شبکه‌ی زنجیره‌ی قالب‌های عمومی و پلتفرم قرارداد هوشمند اتریوم، فایل سیستم همتا به همتای IPFS، و پروتکل چرخ‌دنده دوتایی را معرفی می‌کنیم که نقش پایه‌ای در پروتکل پیشنهادی دارند.

۲-۱- شبکه اتریوم

شبکه‌ی اتریوم به عنوان اولین شبکه زنجیره‌ی قالب‌های نسل دوم دارای قابلیت‌هایی است که امکان توسعه برنامه‌های کاربردی غیرمتمرکز را فراهم می‌سازد [۲۳]. این شبکه دارای نسخه‌های عمومی و خصوصی بوده و وجود ابزاری به نام قرارداد هوشمند و قابلیت‌های آن انعطاف‌پذیری بزرگی را در این سامانه ایجاد کرده است. اتریوم دارای یک زبان اسکریپت نویسی تورینگ-کامل است به گونه‌ای که یک برنامه‌نویس می‌تواند با استفاده از آن قرارداد هوشمند یا تراکنش خود را بسازد [۲۴]. تا به حال زبان‌های سطح بالای متفاوتی برای برنامه‌نویسی قرارداد هوشمند در اتریوم معرفی شده‌اند. از جمله می‌توان به زبان‌های Solidity، Serpent، LLL و Vyper اشاره کرد که از این میان دو زبان Serpent و LLL پیش‌تر منسوخ گشته‌اند. هم‌اکنون، اتریوم از دو زبان Solidity و Vyper پشتیبانی می‌کند. Solidity یک زبان قرارداد-گرا[۹]ست به گونه‌ای که از زبان‌های C++، Python و JavaScript اثر پذیرفته است. هم‌چنین این زبان از ارث‌بری و توابع کتابخانه‌ای پشتیبانی می‌کند.

۲-۲- فایل سیستم همتا به همتای IPFS

در مقاله حاضر برای ذخیره‌سازی کلید عمومی کاربران از فایل سیستم IPFS استفاده می‌شود. IPFS یک سامانه‌ی همتا-به-همتا است [۲۲] که در مقیاس بین‌المللی عمل می‌کند و محدود به یک جغرافیای خاص نیست. هم‌چنین، منبع باز بودن و آدرس‌دهی بر مبنای محتوا از نقاط قوت این سامانه است. آدرس‌دهی به این معناست که IPFS با ذخیره‌سازی یک محتوا یک مقدار درهم یکتا به آن محتوا نظیر می‌کند به گونه‌ای که کاربر با ارائه این مقدار که به آن شناسه محتوا[۱۰] گفته می‌شود، می‌تواند محتوای اصلی را بازیابی کند. این امر تضمین می‌کند که هیچ دو محتوای یکسانی شناسه‌های متفاوت نخواهند داشت [۲۲].

۲-۳- پروتکل چرخ‌دنده دوتایی

در مقاله حاضر برای ایجاد کلیدهای جلسه مورد استفاده در رمزنگاری پیام‌ها از پروتکل چرخ‌دنده دوتایی استفاده شده است. با استفاده از این پروتکل می‌توان پیام‌های ارسالی بین دو طرف یک ارتباط (در تماس‌های صوتی، تصویری و پیام‌رسانی فوری) را با استفاده از یک کلید محرمانه مشترک به صورت امن رمزنگاری کرد و به رمزگذاری انتها به انتها دست یافت. به صورت سنتی، معمولا یک بار توافق یا اشتراک کلید (با استفاده از لفاف‌گذاری کلید یا روش دیفی-هلمن) در ابتدای یک جلسه ارتباطی انجام می‌شود و در فواصل زمانی نسبتا طولانی کلیدها تعویض می‌شود. در حالی که پروتکل چرخ‌دنده دوتایی امنیت اثبات پذیر دارد و در آن با هر بار تبادل پیام، کلیدها تعویض می‌شوند. این امر تحلیل رمز را برای مهاجم بسیار سخت می‌کند. ما برای تفاهم بر سر کلید محرمانه مشترک در چرخ‌دنده دوتایی از دیفی هلمن توسعه‌یافته سه‌گانه (X3DH) استفاده می‌کنیم [۱۲]. در این صورت، ویژگی‌های امنیتی رمزگذاری انتها-به-انتها، احراز هویت، انکار پذیری، امنیت رو به جلو، و امنیت رو به عقب را داریم.

در پروتکل چرخ‌دنده دوتایی این قابلیت فراهم شده است که بتوان برای هر پیام از کلید جدیدی استفاده کرد. این پروتکل را به صورت کلی می‌توان به سه بخش تقسیم کرد که با استناد به [۱۳] تشریح می‌کنیم:

أ. **ایجاد کلید:** این مرحله قبل از ایجاد مکالمه بین طرفین (به عنوان یک قرارداد، از این به بعد دو طرف تبادل کننده پیام را آلیس و باب می‌نامیم) انجام می‌شود و هر طرف بایستی کلیدهای لازم را ایجاد کند.

ب. **توافق کلید:** اولین پیامی که از طرف باب به آلیس ارسال می‌شود و طرفین بر سر کلید مشترک توافق می‌کنند.

ج. **تجدید کلید:** پس از ایجاد مکالمه بین آلیس و باب، به ازای هر پیام کلیدهای جدیدی برای رمزگذاری و رمزگشایی برای هر طرف ایجاد می‌شود.

تابع اشتقاق کلید

تابع اشتقاق کلید (KDF[۱۱]) به عنوان ورودی، یک کلید محرمانه و تصادفی را به همراه یک داده دریافت می‌کند و یک مقدار خروجی که برای استفاده به عنوان کلید رمزنگاری مناسب است را برمی‌گرداند. از روی این خروجی نمی‌توان مقدار کلید ورودی را بدست آورد [۱۲].

چرخ‌دنده کلید متقارن و کلیدهای زنجیره

در این پروتکل، کلید رمزگذاری جدیدی برای هر پیام با استفاده از زنجیره ارسال/دریافت ایجاد، و پس از یکبار استفاده دور انداخته می‌شود. به این کلیدها، کلیدهای زنجیره گفته می‌شود. ورودی‌های KDF در این زنجیره‌ها ثابت هستند و فقط کلید KDF از مرحله قبلی به مرحله بعدی وارد می‌شود. به هر نوبت تولیدِ کلید زنجیره و کلید پیام، یک گام چرخ‌دنده[۱۲] گفته می‌شود.

چرخ‌دنده دیفی‌هلمن

برای شروع به‌کار چرخ‌دنده‌های دیفی‌هلمن، هر کدام از طرفین بایستی یک جفت کلید دیفی‌هلمن تولید کنند که به آن‌ها جفت کلید چرخ‌دنده گفته می‌شود. در هر پیام ارسالی کاربر بایستی به همراه پیام رمز شده قسمت عمومی کلید دیفی هلمن را ارسال کند.

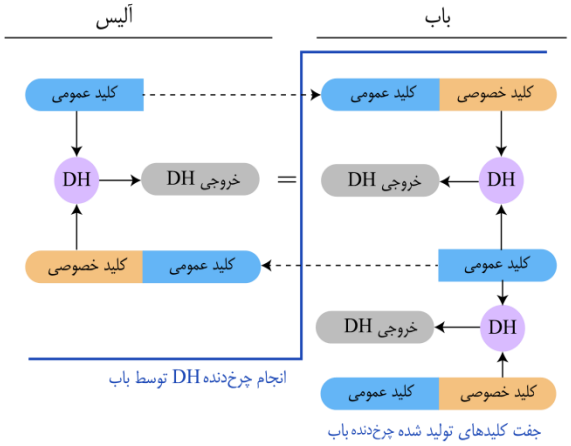
چرخ‌دنده دوتایی

این چرخ‌دنده از ترکیب دو چرخ‌دنده فوق یعنی چرخ‌دنده کلید متقارن و چرخ‌دنده دیفی هلمن بدست می‌آید. در این‌جا ذکر دو نکته لازم است:

أ. در هر بار دریافت کلید عمومی چرخ‌دنده جدید از طرف مقابل گام چرخ‌دنده دیفی هلمن بایستی قبل از گام چرخ‌دنده کلید متقارن انجام شود که در آن دو کلید مشترک جدید تولید می شود؛ یکی برای ایجاد یک زنجیره دریافت جدید و دیگری برای زنجیره ارسال جدید استفاده می‌شود.

ب. وقتی که یک پیام ارسال یا دریافت می شود بایستی گام چرخ‌دنده کلید متقارن انجام شود تا کلیدهای جدید برای ارسال یا دریافت پیام بدست آید.

شکل ۱) فرآیند اولین گام چرخ‌دنده دیفی هلمن در سمت‌های آلیس و باب را نشان می‌دهد.



شکل ۱: اولین گام چرخ‌دنده دیفی هلمن توسط باب (آغاز کننده مکالمه) و آلیس و تولید کلیدهای چرخ‌دنده

چرخ‌دنده دوتایی پساکوانتومی

برای پساکوانتومی کردن پروتکل چرخ‌دنده دوتایی، ما فقط می‌خواهیم قسمت توافق کلید مشترک را پساکوانتومی کنیم. با این‌کار یک الگوریتم پساکوانتومی ترکیبی بدست می‌آید که هم از مزایای رمزنگاری غیرکوانتومی بهره‌مند خواهیم شد و هم از امنیت رمزنگاری پساکوانتومی استفاده می‌شود. در این صورت با توجه به ناشناخته بودن امنیت پساکوانتوم و جدید بودن الگوریتم‌های ارائه شده در این حوزه، در صورتی که الگوریتم‌های پساکوانتومی استفاده شده دچار آسیب‌پذیری امنیتی شده و شکسته شوند، می‌توان جلوی حمله را با استفاده از الگوریتم‌های با سابقه غیرکوانتومی (مانند ECDH و AES) گرفت و اگر با رایانه‌های کوانتومی این الگوریتم‌های غیرکوانتومی شکسته شوند، می‌توان جلوی حملات غیرفعال کوانتومی را با استفاده از الگوریتم‌های پساکوانتومی استفاده شده گرفت.

در بین الگوریتم‌های پساکوانتومی موجود، الگوریتم SIDH مناسب‌ترین گزینه است، چرا که این الگوریتم دارای ساختاری متناسب با ساختار الگوریتم ECDH است و الگوریتم‌های متعدد پساکوانتومی دیفی‌هلمن مانند این الگوریتم به عنوان بدیل ارائه نشده است. شایان ذکر است که الگوریتم SIDH (به طور دقیق‌تر، حالت خاصی از آن که SIKE نامیده می‌شود) در بین الگوریتم‌های نامزد در دور چهارم مسابقات NIST اعلام شده است [۲۰].

الگوریتم SIDH دارای نسخه‌های متعددی است که شامل SIDH434، SIDH503، SIDH610 و SIDH751 بوده که هر کدام دو حالت عادی و فشرده دارند. نسخه‌های مختلف SIDH در واقع سطوح امنیتی مختلفی را فراهم می‌کنند که در جدول ۱) تفاوت طول کلیدهایِ الگوریتم‌های مختلف رمزنگاری استفاده شده در X3DH و سطوح امنیتی آن‌ها آورده شده است. منبع کد استفاده شده در این مقاله برای پیاده‌سازی الگوریتم SIDH توسط شرکت مایکروسافت توسعه داده شده است [۲۵].

جدول ۱: تفاوت طول کلیدهایِ الگوریتم‌های مختلف رمزنگاری استفاده شده در X3DH و سطوح امنیتی آن‌ها

الگوریتم	کلید مشترک (بایت)	کلید عمومی (بایت)	کلید خصوصی (بایت)	سطح امنیت کوانتومی
ECDH	32	32	32	0
SIDH434	110	330	28	1
SIDH503	126	378	32	2
SIDH610	154	462	38	3
SIDH751	188	564	48	5

۳- پروتکل پیشنهادی

پروتکل پیشنهادی در این مقاله بر مبنای ترکیب و بهبود دو روش پیشنهادی در [۱۰ و ۹] است که نتیجه‌ی آن دست‌یابی به یک پروتکل پساکوانتومی است که در همه‌ی سطوح در برابر حملات کوانتومی مقاوم بوده و امکان احراز اصالت موجودیت‌ها را به صورت گمنام فراهم می‌آورد. هم‌چنین، این پروتکل خطر حمله‌ی مرد میانی را برطرف می‌کند. پروتکل متشکل از چهار مرحله اصلی (۱) ثبت‌نام، (۲) دریافت کلید عمومی مخاطب، (۳) دستداد[۱۳]، و (۴) تبادل پیام است. برای روشن‌تر شدن پروتکل در یک مثال فرآیند کلی پروتکل را تشریح می‌کنیم. فرض می‌کنیم آلیس و باب با استفاده از این پروتکل می‌خواهند با یکدیگر ارتباط برقرار کرده و پیامی ارسال کنند. ابتدا آلیس بایستی دو جفت کلید پساکوانتومی با استفاده از الگوریتم SIDH تولید کند. یکی از جفت کلیدها به عنوان کلید بلندمدت هویت استفاده می‌شود و دیگری کلید کوتاه مدت است. برای ثبت کلید هویت در شبکه زنجیره‌ی قالب‌ها ابتدا آلیس کلید عمومی هویت را به کارگزار IPFS ارسال کرده و شناسه آن را دریافت می‌کند، سپس این کلید را به شبکه زنجیره قالب‌ها ارسال کرده و با ایجاد یک قرارداد استعلام و ثبت شناسه IPFS این کلید و شناسه خود در قرارداد این امکان را فراهم می‌کند که دیگران بتوانند با استفاده از قرارداد شناسه کلید آلیس را استعلام بگیرند. سپس آلیس آدرس این قرارداد را به همراه شناسه خود به کارگزار می‌فرستد تا در کارگزار ثبت شود. تا این‌جا مراحل ثبت‌نام آلیس

در پروتکل به پایان رسیده است.

حال فرض کنید باب می‌خواهد به‌صورت امن به آلیس پیام ارسال کند، او برای این کار نیاز دارد کلید عمومی پساکوانتومی آلیس را به‌صورت امن بدست آورد. برای این کار باب شناسه آلیس (فرض می‌کنیم باب شناسه آلیس را دارد) را به کارگزار ارسال کرده و آدرس قرارداد هوشمند و شناسه IPFS کلید عمومی را بدست می‌آورد. باب برای اطمینان از صحت شناسه IPFS آلیس آن را به همراه شناسه خودِ آلیس به قرارداد هوشمند او فرستاده و از درستی آن مطمئن می‌شود. سپس باب از طریق این شناسه کلید عمومی هویت آلیس را از کارگزار IPFS دریافت می‌کند. پس از دریافت کلید عمومی آلیس، باب کلیدهای هویت و یک‌بار مصرف پساکوانتومی خود را برای آلیس ارسال می‌کند (آلیس در این مرحله می‌تواند کلید هویت باب را استعلام بگیرد) و در مرحله بعد آلیس کلید یک‌بار مصرف پساکوانتومی و کلید یک‌بار مصرف کلاسیک خود را برای باب ارسال می‌کند. با استفاده از پروتکل X3DH طرفین می‌توانند یک کلید ریشه پساکوانتومی تولید کنند. این کلید پساکوانتومی به عنوان کلید اولیه در پروتکل دیفی‌هلمن استفاده می‌شود و کلیدهای بعدی کلاسیک با استفاده از این کلید ریشه‌ی پساکوانتومی بدست می‌آیند. در ادامه و با جلو رفتن مراحل پروتکل دیفی‌هلمن طرفین ارتباط به یک ارتباط رمزشده پساکوانتومی ترکیبی انتها-به-انتها دست میابند.

شایان ذکر است که اگر کلید هویت کاربری منقضی یا باطل شود، لازم است برای از سرگیری تبادل پیام، تمام مراحل از اول اجرا شود. اما در غیر این صورت، تبادلات در مرحله آخر باقی می‌مانند و نیازی به تکرار مراحل از ابتدا نیست. در ادامه به تشریح جزئیات پروتکل پیشنهادی می‌پردازیم.

۳-۱- موجودیت‌های درگیر

پیش از تشریح پروتکل‌های مربوط به سامانه، موجودیت‌های درگیر در آن و عملیات در نظر گرفته شده را تعریف می‌کنیم:

- **کاربر**: موجودیتی است که یک طرف مکالمه در پروتکل پیشنهادی است. کاربران یک حساب کیف پول بر روی شبکه‌ی اتریوم دارند و حداقل یک جفت کلید چرخدنده دوتایی برای خود ساخته‌اند. به گونه‌ای که قرارداد هوشمندِ این موجودیت شناسه‌ی IPFS کلید عمومی این موجودیت را نگهداری کند. چنین موجودیتی دارای یک شناسه کاربری است و رکوردهای دارای این شناسه، به این موجودیت منتسب می‌شوند.
- **کارگزار**: موجودیتی است که یک پایگاه داده از برخی اطلاعات کاربر دارد و آخرین وضعیت مخاطبین پیش از برقراری ارتباط از کارگزار استعلام می‌شود. رکورد هر کاربر در خود شناسه‌ی کاربر، نشانی آخرین قرارداد هوشمند مستقر شده[۱۴]ی تحت مالکیت آن کاربر، نشانی کیف پول آن کاربر، شناسه‌ی IPFS کلید عمومی وی و تاریخ انقضای کلید را ذخیره می‌کند. کارگزارها هم یک حساب کیف پول اتریوم دارند.
- **قرارداد هوشـــمند:** وظیفه نگهداری اطلاعات به صـــورت غیر قابل دستکاری را بر عهده دارد.

۳-۲- مرحله ثبت نام

ساده‌ترین حالت را در نظر می‌گیریم که در آن این سامانه از سه موجودیت تشکیل شده است: آلیس، باب و کارگزار که همگی دارای حساب در زنجیره‌ی قالب‌های عمومی اتریوم هستند. فرض کنیم آلیس دارای شناسه کاربری یکتایی است و یک جفت کلید پساکوانتومی SIDH برای خود ایجاد کرده است. این کلید نسبت به سایر کلیدها در پروتکل عمری طولانی‌تر دارد و کلید هویت نامیده می‌شود. او کلید عمومی هویت خود را در IPFS ذخیره کرده (گام۱) و شناسه‌ی IPFS این کلید را دریافت می‌کند (گام۲). به عنوان یک قرارداد، از این پس در این متن، شناسه‌ی محتوای کلید عمومی چرخدنده دوتایی کاربر را "شناسه‌ی IPFS کلید عمومی" آن کاربر می‌نامیم. سپس یک قرارداد هوشمند می‌سازد و شناسه‌ی IPFS کلید عمومی را به همراه شناسه‌ی کاربری خود و مقدار دستمزدی که استعلام‌گیرنده باید بپردازد (بعداً تشریح می‌شود)، در قرارداد هوشمند ذخیره می‌کند (گام۳). استقرار قرارداد در قالب یک تراکنش درون یک قالب از زنجیره‌ی قالب‌ها ثبت می‌شود. برای امنیت بیشتر و پرهیز از استفاده طولانی مدت از یک جفت کلید، قرارداد هوشمند طوری طراحی شده است که فقط تا مدت محدودی قابل استفاده باشد (مثلاً ۱۸۰ روز) و پس از آن تراکنش‌های استعلام (مرحله بعد) برگشت[۱۵] داده می‌شوند.

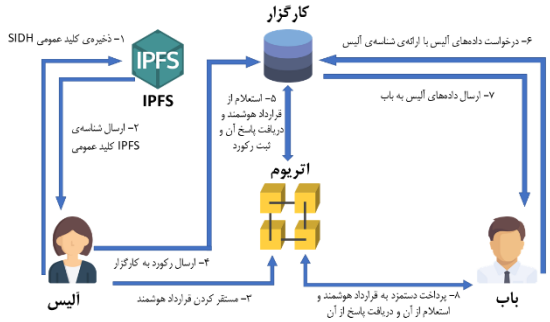
در این مرحله، آلیس شناسه‌ی کاربری خود، شناسه‌ی IPFS کلید عمومی خود، نشانی قرارداد هوشمند مستقر شده، و نشانی کیف پول خود را به کارگزار می‌فرستد تا آن‌ها را در قالب یک رکورد در پایگاه داده خود ذخیره کند (گام۴).

کارگزار پیش از ذخیره این رکورد، به قرارداد هوشمند مستقر شده توسط آلیس متصل شده و تطابق شناسه‌ی IPFS و کلید عمومی را از آن استعلام می‌گیرد. قرارداد هوشمند، با استفاده از شناسه IPFS کلید عمومی، که آلیس در آن ذخیره کرده است، یکی از دو پاسخ "کلید عمومی مورد تأیید است" یا "کلید عمومی مورد تأیید نیست" را به سمت کارگزار برمی‌گرداند و این پاسخ را در دفتر کل ثبت می‌کند (گام ۵). اگر کارگزار پاسخ تأیید را دریافت کند، رکورد مورد نظر را در پایگاه داده‌ی خود ذخیره خواهد کرد و در غیر این صورت، این رکورد را ذخیره نخواهد کرد.

۳-۳- مرحله دریافت کلید عمومی مخاطب

فرض کنیم باب نیز تمام مراحل فوق را طی کرده است. اکنون باب نیاز دارد تا کلید عمومی ارائه شده توسط آلیس را اصالت سنجی کند (شکل ۲)). او با ارائه‌ی شناسه آلیس به کارگزار، موارد ذیل را از او درخواست می‌کند: نشانی قرارداد هوشمند و کیف پول آلیس، شناسه‌ی IPFS کلید عمومی آلیس و نشانی کیف پول خودِ کارگزار (گام‌های ۶ و ۷).

باب به قرارداد هوشمند آلیس متصل شده و از آن استعلام می‌گیرد. به همراه این استعلام دستمزد قرارداد هوشمند نیز ارسال می‌شود (گام ۸). مقدار این دستمزد پیش‌تر و در طی تشکیل سامانه، توافق شده است. این دستمزد برای دریافت خدمت استعلام، به قرارداد داده می‌شود به گونه‌ای که اگر باب دستمزد مورد نظر را پرداخت نکند اجازه‌ی استعلام گرفتن از قرارداد هوشمند را نخواهد داشت. باب در استعلام خود تمام داده‌های دریافتی از کارگزار خصوصاً اصالت شناسه‌ی IPFS کلید عمومی را از قرارداد می‌پرسد.



شکل ۲: روند کاری مراحل ثبت نام آلیس (سمت چپ) و دریافت کلید عمومی آلیس توسط باب (سمت راست)

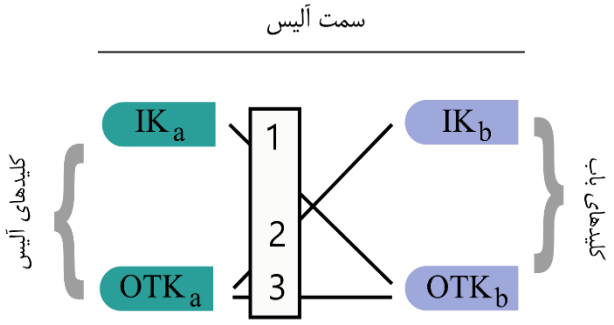
پاسخ استعلام می‌تواند یکی از این موارد چهارگانه باشد: "اصالت کلید عمومی توسط کارگزار استعلام گرفته شده است و اصالت آن مورد تأیید است"، "اصالت کلید عمومی توسط کارگزار استعلام گرفته شده است و اصالت آن مورد

تأیید نیست"، "کاربر مورد نظر مالک این قرارداد هوشمند نیست و ارتباطی با آن ندارد" یا "کارگزار مورد نظر پیش‌تر استعلامی نگرفته است". تنها پاسخ نخست ادعای آلیس را تصدیق می‌کند و نشان می‌دهد که کلید عمومی ارائه شده توسط آلیس دارای اصالت است. پاسخ‌های دوم و سوم گواهی بر کذب بودن ادعای آلیس هستند. هم‌چنین، پاسخ چهارم گواهی بر کذب بودن ادعای کارگزار است. باب با توجه به پاسخی که دریافت می‌کند درباره‌ی ادامه تعامل با آلیس تصمیم‌گیری خواهد کرد.

۳-۴- مرحله دستداد

مرحله توافق کلید اولیه در پروتکل پیشنهادی با روش دیفی هلمن توسعه‌یافته سه‌گانه (X3DH) انجام می‌شود. در پروتکل چرخ‌دنده دوتایی عادی برای این کار از الگوریتم ECDH استفاده شده است [۱۳]؛ اما ما برای مقاوم‌سازی نسبت به حملات کوانتومی، از الگوریتم SIDH استفاده می‌کنیم. از کلید توافق شده در این مرحله به عنوان کلید ریشه در مرحله تبادل پیام (با استفاده از چرخ‌دنده دوتایی) استفاده می‌شود.

نحوه انجام پروتکل X3DH در شکل ۳) نشان داده شده است. نکته مهمی که در پروتکل پیشنهادی در نظر گرفته شده است این است که کارگزار در اینجا نسبت به پروتکل چرخ‌دنده دوتایی عادی نقش کم‌تری دارد؛ چرا که در غیر این‌صورت برای مقابله با حملات مرد میانی بایستی برای هر کلید یک‌بار مصرف نیز از قرارداد هوشمند استفاده می‌شد. در این صورت هزینه فرآیند برای مالک کلید و مخاطبان وی افزایش می‌یافت و سربار بیشتری هم داشت که کارایی پروتکل را کاهش می‌داد.



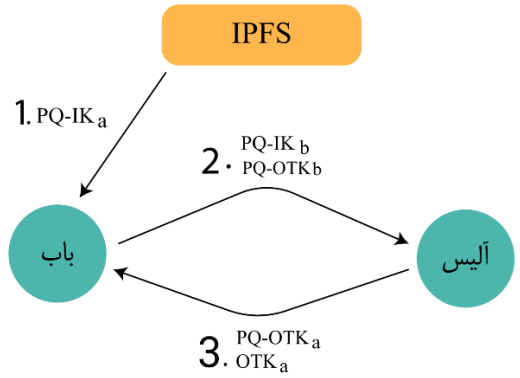
شکل ۳: نحوه انجام پروتکل X3DH در سمت آلیس

بنابراین باب فقط کلید عمومی هویت آلیس را تنها یک‌بار در مرحله قبل از کارگزار دریافت می‌کند و با کمک قرارداد هوشمند اصالت‌سنجی می‌کند؛ و دیگر با کارگزار و قرارداد هوشمند تعاملی ندارد. کلیدهای دیگر مورد نیاز برای انجام پروتکل X3DH مستقیماً توسط آلیس و باب برای یکدیگر ارسال می‌شوند.

مراحل دقیق دستداد در شکل ۴) نشان، و در ادامه شرح داده شده است (در همه موارد منظور از کلید، قسمت عمومی کلید است.) در ابتدا باب به‌عنوان آغازگر ارتباط، کلید هویت پساکوانتومی SIDH آلیس ($PQ\text{-}IK_a$) را از آدرس IPFS ذکر شده توسط کارگزار در مرحله قبل می‌گیرد (گام۱). سپس کلید هویت پساکوانتومی خود ($PQ\text{-}IK_b$) و یک کلید یک‌بار مصرف ($PQ\text{-}OTK_b$) را برای آلیس ارسال می‌کند (گام ۲).

آلیس با همان روش توضیح داده شده در مرحله قبل (دریافت کلید عمومی مخاطب) از صحت کلید مطمئن می‌شود. سپس یک کلید یکبار مصرف پساکوانتومی ($PQ\text{-}OTK_a$) را به باب ارسال می‌کند (گام ۳)؛ مقدار دیگری که در این گام ارسال می‌شود در مرحله بعد تشریح شده است.) در پایان این مرحله باب و آلیس با استفاده از قسمت خصوصی کلیدهای خود و کلیدهای دریافتی از طرف مقابل و طبق پروتکل X3DH با استفاده از SIDH در خصوص یک کلید مشترک به تفاهم می‌رسند. این کلید به یک KDF داده می‌شود تا از تشخیص‌ناپذیری بهتری برخوردار باشد. خروجی این KDF در واقع کلید ریشه اولیه پروتکل چرخ‌دنده دوتایی است.

شایان ذکر است که کلیدهای رد و بدل شده در این مرحله توسط پروتکل رمزگذاری سرآیند چرخ‌دنده دوتایی به صورت تماماً رمز شده در شبکه منتقل می‌شوند.



شکل ۴: فرآیند تبادل کلیدها در مرحله دستداد

۳-۵- مرحله تبادل پیام

تا به اینجا، کلیدهای هویت و چهار کلید PQ رد و بدل شده در مرحله دستداد همه از نوع پساکوانتومی بوده‌اند (SIDH) و پروتکل را نسبت به حملات کوانتومی مقاوم کرده‌اند. اما برای کاهش اندازه پیام‌ها و سربارهای رمزنگاری، کلیدها و عملیات مرحله تبادل پیام را از نوع غیرکوانتومی در نظر گرفته‌ایم. به این منظور در گام ۳ از مرحله دستداد، آلیس یک جفت کلید ECDH (غیرکوانتومی) هم ایجاد می‌کند و قسمت عمومی آن (OTK_a) را به باب ارسال می‌کند. از این مقدار در آغاز مرحله تبادل پیام استفاده می‌شود.

باب هم در آغاز این مرحله یک جفت کلید ECDH ایجاد می‌کند و قسمت عمومی آن (OTK_b) را در اولین پیام به آلیس می‌فرستد. طرفین با این کلیدها گام چرخ‌دنده دیفی هلمن را اجرا کرده و با استفاده از کلید مشترک دیفی‌هلمن بدست آمده در این مرحله و کلید ریشه اولیه بدست آمده در مرحله دستداد (کلید حاصل از X3DH) و دادن آن‌ها به یک KDF، کلید ریشه پروتکل چرخ‌دنده دوتایی را به دست می‌آورند. سپس با استفاده از چرخ‌دنده کلید متقارن، زنجیره‌های ارسال و دریافت و کلیدهای مرتبط با هر کدام را می‌سازند. به این ترتیب تولید کلید و تبادل پیام امن طبق پروتکل چرخ‌دنده دوتایی ادامه می‌یابد.

۳-۶- فاز قرارداد هوشمند

زنجیره‌ی قالب‌ها دارای دو ابزار قدرتمند است که از آن‌ها در اصالت سنجی کلید هویت کاربران استفاده شده است: ۱) زنجیره‌ای از قالب‌ها. ۲) قرارداد هوشمند. زنجیره‌ی قالب‌ها دارای ویژگی برجسته تغییرناپذیری است. به این معنا که داده‌های ثبت شده در زنجیره قابل تغییر یا حذف نیستند. این ویژگی به ما اطمینان می‌دهد که شناسه‌ی IPFS کلید عمومی که در فرآیند اعتبارسنجی مورد استفاده قرار می‌گیرد، هرگز دستخوش تغییر نمی‌شود و دقیقا دارای همان مقداری است که توسط مالک قرارداد ثبت شده است. از طرف دیگر، قرارداد هوشمند با اجرای خودکار تمام دستورها و همواره در دسترس و در حال اجرا بودن و تغییرناپذیری کدی که حاوی آن است، این اطمینان را به

تمام طرف‌های درگیر می‌دهد که وظیفه از پیش توافق شده به طور صحیح و کامل انجام می‌پذیرد.

در ادامه این بخش، به بیان عملکرد قرارداد هوشمند می‌پردازیم. سپس توضیح می‌دهیم چه اســـتفاده‌ای از فضـــای ذخیره‌ســـازی که زنجیره‌ی قالب‌ها به ما می‌دهد، می‌کنیم و چه داده‌هایی را در آن ذخیره می‌کنیم.

عملکرد و نقش قرارداد هوشمند

به طور کلی، قرارداد هوشمند کاربر امکان اصالت‌سنجی کلید هویت کاربر را برای استعلام گیرنده فراهم می‌آورد. کد قرارداد هوشمند شامل توابع ذیل است:

createContract(): نخســـتین تابعی اســـت که پس از مســـتقر شـــدن قرارداد هوشمند و فقط توسط مالک قرارداد اجرا می‌شود و شناسه IPFS کلید عمومی مالک و تاریخ انقضا را ثبت می‌کند. هم‌چنین، در این تابع مقدار دستمزدی که موجودیت استعلام گیرنده باید به قرارداد هوشمند بپردازد تعیین می‌شود. این تابع به گونه‌ای نوشته شده است که دقیقا یک بار اجرا شود. یعنی پس از اولین اجرای موفقِ آن، تمام تراکنش‌هایی که شـــامل تعامل با این تابع هســـتند برگشت داده خواهند شد.

requestApproval(): تابع استعلام گرفتن کارگزار درباره‌ی اصالت کلید عمومی مالک قرارداد هوشمند است.

traceBackHistory(): اگر آلیس بخواهد کلید هویت باب را اصـــالت ســـنجی نماید، به قرارداد هوشمند باب متصل شده و این تابع را اجرا می‌کند.آلیس باید همزمان با فراخوانی این تابع دستمزد استعلام را هم به صورت اتر ارسال کند.

interactionView(): این تابع مجموع تعداد استعلام‌های کارگزارهای مختلف و مجموع تعداد اســـتعلام‌های موفق، که پاســـخ آن‌ها تأیید اصـــالت کلید عمومی است را برمی‌گرداند. این تابع از دسته‌ی "توابع مقدار-گیرنده[۱۶]" است و اجرای آن هیچ هزینه‌ای ندارد و گاز مصرف نمی‌کند.

withdraw(): پس از تجمیع دســـتمزدها در این قرارداد هوشـــمند، مقداری از آن‌ها باید به حســـاب موجودیت ذی‌نفع، که پیش از راه‌اندازی ســـامانه تعریف می‌شود، واریز شود. *withdraw()* این انتقال را از قرارداد هوشمند به آن حساب انجام می‌دهد. نشانی گیرنده این دستمزدها در بدنه‌ی *withdraw()* حک شده است.

destruct(): اجرای این تابع قرارداد را غیرفعال می‌کند. به عبارت دیگر، هیچ تعاملی با این قرارداد هوشمند قابل اجرا نیست. قرارداد هوشمند به گونه‌ای نوشته شده است که با فرا رسیدن تاریخ انقضای آن این تابع به طور خودکار اجرا شود. این تابع، در ابتدا تمام موجودی قرارداد را به نشانیِ از پیش تعیین شده‌ای فرستاده سپس قرارداد را غیر فعال می‌کند.

ثبت داده در دفترکل

قرارداد هوشمند به گونه‌ای نوشته شده است که دسته‌ای از داده‌ها را در دفتر کل ثبت می‌کند. این داده‌ها عبارت از شناسه‌ی کاربر مالک قرارداد هوشمند، نشانی کیف پول کاربر مالک قرارداد هوشمند، شناسه‌ی IPFS کلید عمومی کاربر مالک، مجموع تعداد استعلام‌های کارگزار، مجموع تعداد استعلام‌های با پاسخ تأیید اصالت، تاریخ انقضای قرارداد هوشمند، یک ذخیره‌گاه کلید-مقدار یا نگاشت (که کلید آن نشانی کیف پول کارگزار و مقدار آن یک مقدار دودویی است: به گونه‌ای که اگر پاسخ استعلام کارگزار از قرارداد هوشمند تأیید اصالت باشد مقدار دودویی true و در غیر این صورت false خواهد بود)، نتیجه‌ی استعلام کاربر از قرارداد هوشمند (هر کدام از این نتایج در قالب یک رویداد در دفتر کل ثبت می‌شوند)، رویداد ایجاد قرارداد (که با تعیین نشانی کیف پول مالک آن در دفتر کل ثبت می‌شود)، رویداد استعلام اصالت کلید عمومی از سوی کارگزار (که اعلام می‌کند کارگزاری از قرارداد هوشمند در خواست کرده است تا کلید عمومی را اعتبارسنجی کند.) هستند.

۴- پیاده‌سازی و ارزیابی کارایی

ما در این بخش، در ابتدا پیاده‌سازی الگوریتم X3DH با SIDH و پروتکل چرخ‌دنده دوتایی را که مربوط به مراحل دستداد و تبادل پیام هستند توضیح می‌دهیم. پس از آن، به توضیح پیاده‌سازی مراحل ثبت نام و دریافت کلید عمومی مخاطب در سطح زنجیره‌ی قالب‌ها می‌پردازیم. در این توضیح به معرفی ابزارها و محیط‌های به کار رفته پرداخته، مصرف گاز را تحلیل کرده، کار خود را با نمونه‌ی ارائه شده در [۲۲] مقایسه کرده، و در پایان به تحلیل امنیتی آن می‌پردازیم.

۴-۱- پیاده‌سازی و ارزیابی دستداد و تبادل پیام

در این مقاله بر خلاف [۲۱] از ترکیبی از توافق کلید پساکوانتومی و غیر آن استفاده شده است و همه توافق کلیدها پساکوانتومی نیستند. به همین دلیل سربار پروتکل چرخ‌دنده دوتایی به نسبت بسیار کاهش یافته است. پیاده‌سازی الگوریتم SIDH با استفاده از زبان C انجام شده است. بقیه قسمت‌ها به زبان Python هستند. برای ذخیره‌سازی اطلاعات کلید کاربران در سمت کارگزار نیز از پایگاه داده MySQL استفاده شده است.

ارزیابی دستداد و تبادل پیام به صورت جداگانه انجام شده است. در مرحله دستداد ایجاد دو جفت کلیدِ پساکوانتومیِ هویت و یک‌بار مصرف برای طرفین ارتباط را داریم که پیام‌ها مطابق شکل ۴) است و محاسبات طبق پروتکل X3DH برای بدست آوردن کلید ابتدایی ریشه است. ارزیابی مرحله تبادل پیام نیز شامل یک حلقه nتایی است که در آن باب (فرستنده) یک پیام را رمز کرده و برای آلیس (گیرنده) ارسال می‌کند. آلیس پیام را رمزگشایی می‌کند و با پیام اصلی باب مقایسه می‌کند. سپس آلیس پیام جدیدی را تولید می‌کند، آن را رمز می‌کند و برای باب ارسال می‌کند. باب نیز پیام را دریافت کرده، رمزگشایی می‌کند و با پیام اصلی آلیس مقایسه می‌کند. در واقع در هر حلقه دو بار عملیات رمزنگاری و رمزگشایی توسط آلیس و باب انجام می‌شود. این آزمایش ۲۵ بار در حالت‌های یکسان و برای nهای مختلف شامل ۱۰، ۱۰۰ و ۱۰۰۰ با استفاده از رایانه‌ای با مشخصات زیر انجام شده است:

16GB DDR3 RAM (1600Mhz), CPU i5-3230M, 2.60 GHz (2 Cores, 4 Logical Processors)

سپس از نتایج بدست آمده از این دو نوع آزمایش (۲۵ آزمایش) میانگین گرفته شده و خروجی آن‌ها ترسیم شده است. شکل‌های ذکر شده امکان مقایسه بین زمان مرحله دستداد (پساکوانتومی) و تبادل پیام را فراهم می‌کند. تفاوت شکل‌ها در تعداد پیام‌های رد و بدل شده است به طوریکه در شکل ۵) به ترتیب میانگین زمان برای ۱۰ پیام، ۱۰۰ پیام، و ۱۰۰۰ پیام را می‌بینیم. محور افقی شکل‌ها حالات مختلف الگوریتم SIDH را با امنیت پساکوانتومی متفاوت نشان می‌دهد (به جدول ۱) توجه فرمایید.)

نکته‌ای که در این آزمایش‌ها قابل توجه است این است که هر چه تعداد پیام‌های ارسال شده در یک نشست و در مرحله دوم افزایش یابد، سربار ناشی از مرحله ابتدایی که شـــامل تولید کلیدهای پســـاکوانتومی و انجام پروتکل X3DH است به نسبت کم می شود تا جایی که در حالت n=1000 این زمان به‌صـــورت میانگین کم‌تر از زمان رمزنگاری و رمزگشـــایی پیام توســـط آلیس و باب می‌شـــود. دقت شـــود که در این ارزیابی زمان مرحله ابتدایی به‌صـــورت میانگین آورده شـــده اســـت. یعنی زمان کل این مرحله که تنها یک بار در ابتدا انجام می شود بر n تقسیم شده است. همانطور که مشاهده می شود متوسط زمان تولید پیام نزدیک به ۳ میلی‌ثانیه است. در حالی که دستداد با سریع‌ترین

حالت پساکوانتومی فقط ۳۴۳ میلی‌ثانیه زمان می‌برد. همچنین استفاده از حالت فشرده اثر قابل ملاحظه‌ای بر افزایش سربار دارد.

مقایسه دیگری که انجام شده است، مقایسه زمان کامل دو مرحله در حالت استفاده از الگوریتم پساکوانتومی SIDH در حالات مختلف با الگوریتم ECDHE در حالت Curve25519 است که در شکل ۶) آمده است. نکته‌ای که در این‌جا نیز مشاهده می‌شود این است که با افزایش تعداد پیام‌های رد و بدل شده در یک نشست، سربار فاز پساکوانتومیِ ابتدایی کم شده و زمان کامل انجام فرآیند در حالت پساکوانتومی تفاوت زیادی با حالت غیرکوانتومی ندارد.

۴-۲- پیاده‌سازی و ارزیابی قرارداد هوشمند

برای پیاده‌سازی در سطح شبکه زنجیره قالب از نمونه عمومی اتریوم استفاده می‌کنیم. شبکه عمومی اتریوم به علت غیرمتمرکز بودن، در دسترس بودن برای همه، حفظ گمنامی موجودیت‌ها در ارتباط با یکدیگر و امکان کار با قراردادهای هوشمند، امکانات کافی را در اختیار ما می‌گذارد. در ادامه جزئیات پیاده‌سازی آمده است.

بهبودهای کد پیشنهادی نسبت به کار قبلی

کارگزار و قرارداد هوشمند در این مقاله نسخه بهبودیافته نسبت به [۲۲] هستند. به طور خاص کارگزار دخالت کمتری در فرایندها دارد و اطلاعات کمتری را هم نگه می‌دارد. فضای ذخیره‌سازی در زنجیره‌ی قالب‌ها، مصرف گاز، و تعداد تراکنش‌های لازم برای انجام کارها در قرارداد هوشمند کم‌تر شده و برخی نقاط ضعف امنیتی در کد قرارداد برطرف شده است.

در مقاله حاضر، کارگزار هیچ کلید یک‌بار مصرفی را نگهداری نمی‌کند. در حالی که در [۲۲] کارگزار باید برای هر درخواست برقراری ارتباط از سوی باب، یک کلید (عمومی) یک‌بار مصرف جدید از آلیس داشته باشد. به این منظور هر کاربر تعداد مناسبی (مثلا ۱۰۰ عدد) کلید یک‌بار مصرف را به کارگزار ارسال می‌کند تا ذخیره کافی داشته باشد. وقتی کلیدهای یک‌بار مصرف کاربری در حال تمام شدن است، باید کلیدهای یک‌بار مصرف خود را مجدداً تکمیل کند. مدیریت این امر یک سربار ذخیره‌سازی و مدیریت کلید دارد؛ در حالی که کلیدهای یک‌بار مصرف در روش پیشنهادی در مرحله دستداد و به صورت رمزشده بین طرفین ارتباط تبادل می‌شود.

در جدول ۲) مقدار مصرف گاز و تعداد تراکنش‌های ارسالی از سوی هر موجودیت را برای یک بار اجرای پروتکل نشان می‌دهد. همان‌طور که در این جدول مشاهده می‌شود، میزان مصرف گاز مالک و کارگزار به ترتیب ۹۳٪ و ۴۸٪ کاهش یافته است؛ اما گاز مصرفی استعلام گیرنده به علت پرداخت دستمزد نسبت به [۲۲] (فاقد ساز و کار دستمزد) ۴۶٪ افزایش یافته است. یکی از دلایل کاهش گاز آن است که در کد حاضر فضای ذخیره‌سازی کلید- مقدار از ۴ نگاشت به ۲ نگاشت کاهش یافته است.

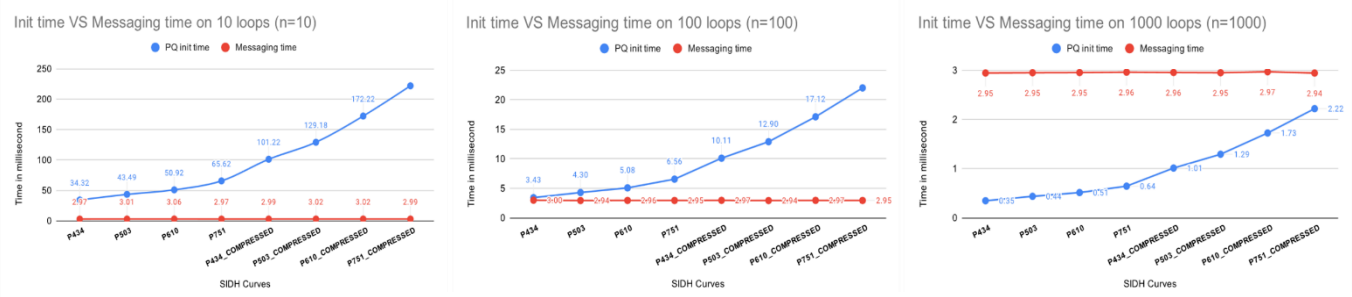
در مرجع [۲۲] تراکنش‌های متعددی را با قرارداد هوشمند داریم. هر کاربر بایستی برای کلید هویت خود یک قرارداد هوشمند ایجاد کند تا کلیدها در دفتر کل ثبت شود. پس از آن اطلاعات این کلیدها را به سمت کارگزار ارسال می‌کند تا در کارگزار هم ذخیره شود. کاربر استعلام گیرنده نیز جهت احراز هویت کاربر مورد نظر خود، باید کلیدِ هویت را اصالت سنجی کند (این کلید به سمت قرارداد هوشمند فرستاده شده و قرارداد هوشمند عمل وارسی را انجام می‌دهد.) در نتیجه، برای هر بار احراز هویت، قرارداد هوشمند باید کلید را مورد وارسی قرار دهد و یک تراکنش در دفتر کل ثبت شود. این موارد مقدار مصرف گاز را افزایش می‌دهد که در طرح بهبودیافته‌ی ما، فقط یکبار تا فرا رسیدن تاریخ انقضای کلید هویت، استعلام‌گیری از قرارداد هوشمند انجام می‌شود.

جدول ۲: مقایسه‌ی عملکرد قرارداد هوشمند در تعداد تراکنش‌ها و مصرف گاز.

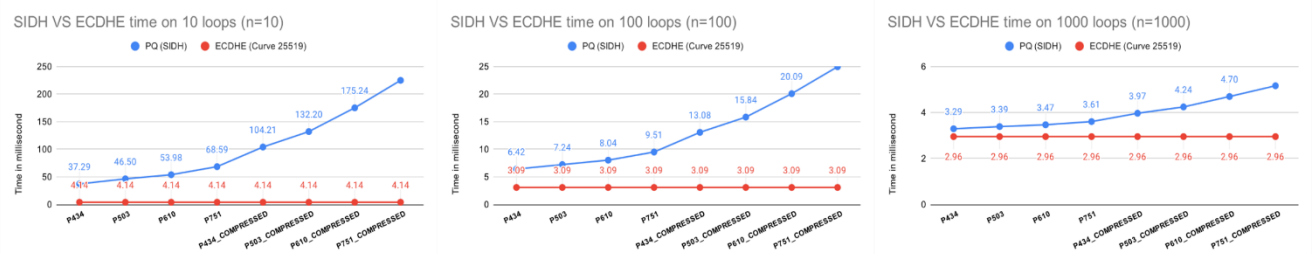
موجودیت	مصرف گاز در [۲۲]	مصرف گاز حاضر	تعداد تراکنش‌ها در [۲۲]	تعداد تراکنش‌های حاضر
مالک	۱۹۰۷۷۳۲ یا ۱۹۶۰۸۳۰	۹۹۰۸۶۷	۳	۲
کارگزار	۱۳۶۵۰۸	۹۲۱۳۳ یا ۴۱۸۹۰	۱	۱
استعلام گیرنده	۲۹۷۹۱	بین ۶۳۴۸۵ تا ۶۵۰۶۱	۱	۲

در کد به کار رفته در [۲۲] فرآیند استعلام‌گیری به طور خودکار انجام نمی‌شود و نیازمند در دسترس بودن کاربرِ مورد وارسی و اجرای بخشی از کد توسط اوست که در صورت در دسترس نبودن آن کاربر امکان به سرانجام رسیدن استعلام وجود ندارد. همچنین، قرارداد هوشمندی که در [۲۲] ارائه شده است پس از یک بار احراز هویت توسط یک کاربر متقاضی، امکان به‌روزرسانی کلید کاربر مورد تقاضا را ندارد. علاوه بر این، کد حاضر قابلیت دریافت و مدیریت دستمزد را نیز دارد.

در کد قرارداد هوشمند بهبودیافته، یک موجودیت مجاز به اجرای هر تابع دلخواهی نیست و کنترل دسترسی انجام می‌شود. بدین گونه که توابع پیش از انجام کارها بررسی می‌کنند که آیا فراخواننده مجاز به اجرا است یا خیر. البته دو تابع *destruct()* و *withdraw()* فاقد کنترل دسترسی هستند زیرا *destruct()* پس از سررسید ۱۸۰ روزه و به طور خودکار اجرا خواهد شد. همچنین موجودیت‌های غیرمجاز انگیزه‌ای برای اجرا کردن *withdraw()* ندارند زیرا نشانی گیرنده در کد آن حک شده است و مهاجم نمی‌تواند موجودی حساب قرارداد هوشمند را برداشت کند.



شکل ۵: مقایسه زمان ایجاد کلیدهای اولیه با زمان ارسال پیام در حالت *n=10, n=100, n=1000*



شکل ۶: مقایسه زمان کامل ارسال پیام در حالت کوانتومی و غیرکوانتومی با *n=10, n=100, n=1000*

استفاده از تحلیلگر Mythril

تحلیلگر امنیتی Mythril مبتنی بر تحلیل کانکولیک[۱۷]، تحلیل لکه[۱۸] و بررسی کردن روند نظارتی بر کد-بایتی ماشین مجازی اتریوم، جهت حذف اضافات فضای جستجو و جهت یافتن مقادیری است که اجازه بهره‌بری از آسیب پذیری های موجود در قرارداد را می‌دهند [۲۶]. مجموعه‌ی تمام آسیب‌پذیری‌هایی که Mythril می‌تواند آن‌ها را تشخیص دهد در [۲۷] قابل مشاهده است.

عملکرد ۹ تحلیلگر مختلف امنیتی قرارداد هوشمند در [۲۶] و [۲۸] با یکدیگر مقایسه شده است. در هر دو مورد، تحلیلگر Mythril توانسته است دقت بیشتری را در تشخیص آسیب‌پذیری‌های موجود در قرارداد هوشمند داشته باشد. با توجه به این موضوع، برای تحلیل امنیتی قرارداد هوشمند خود از Mythril استفاده کرده‌ایم و مسائل احتمالی موجود آن را برطرف نموده‌ایم.

۵- جمع‌بندی و نتیجه‌گیری

در مقاله حاضر از پروتکل رمزنگاری پساکوانتومی چرخ‌دنده دوتایی و شبکه اتریوم استفاده کرده‌ایم. با استفاده از این پروتکل رمزنگاری هر کاربر دارای دو جفت کلید می‌شود به گونه‌ای که بخش کلید عمومی آن‌ها جهت احراز هویت استفاده خواهند شد. قرارداد هوشمند عمل وارسی کلید را انجام داده و احراز هویت را به سرانجام می‌رساند. در این پروتکل، گمنامی و حفظ حریم خصوصی تمام موجودیت‌ها در کل فرآیند احراز هویت حفظ شده و نتیجه احراز هویت قابل اطمینان است. به عنوان کارهای آینده می‌توان به پیاده‌سازی الگوریتم‌های پساکوانتومی دیگر در فاز تبادل کلید و افزودن قابلیت انتخاب الگوریتم در قرارداد هوشمند اشاره کرد. همچنین در زمینه دریافت کارمزد در روش پیشنهادی می‌توان به راهکارهایی جهت ساده‌تر کردن دریافت هزینه‌ها به صورت ارز رایج (غیر رمزارز) با استفاده از یک گره تسهیلگر پرداخت تا کاربرد پروتکل پیشنهادی برای عموم ساده‌تر باشد.

مراجع

[۱] زینب اسکندری، مرجان کائدی، علی بهلولی، «استخراج توکن‌های رمزنگاری جستجوپذیر از ترافیک فشرده‌شده HTTPS به‌منظور بازرسی محتوایی»، مجله مهندسی برق دانشگاه تبریز، جلد ۵۰، شماره ۳، صص. ۱۰۱۱-۱۰۲۳، ۱۳۹۹.

[۲] مهرداد زبیری، بابک مظلوم نژاد میبدی، «معرفی روش جدید رمزنگاری مبتنی بر تولید متن رمز شده متغیر»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۹، شماره ۲، صص. ۶۴۴-۶۲۷، ۱۳۹۸.

[3] L. Chen, et al., “Report on Post-Quantum Cryptography,” Report NISTIR 8105, National Institute of Standards and Technology, US, 2016, Available online at: https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf [Accessed July 16, 2022].

[4] “Post-Quantum Cryptography: Call for Proposals,” National Institute of Standards and Technology, US, July 7, 2022, Available online at: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals [Accessed July 16, 2022].

[5] PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates, National Institute of Standards and Technology, US, July 05, 2022, Available online at: https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4 [Accessed July 16, 2022].

[6] T. M. Fernández-Caramès, P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," IEEE Access, vol. 8, pp. 21091-21116, 2020.

[7] N. Bindel, U. Herath, M. McKague, D. Stebila, "Transitioning to a quantum-resistant public key infrastructure," International Workshop on Post-Quantum Cryptography, pp. 384-405, 2017.

[8] P. Kampanakis, P. Panburana, E. Daw, D. Van Geest, "The Viability of Post-quantum X. 509 Certificates," IACR Cryptol. ePrint Arch., vol. 2018, p. 63, 2018.

[9] M. Braithwaite, “Experimenting with Post-Quantum Cryptography,” Google online security blog, July 2016, Available online at: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html [Accessed on July 17, 2021].

[10] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe, "Post-quantum key exchange—a new hope," 25th USENIX security symposium (USENIX security 16), pp. 327–343, 2016.

[11] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, D. Stebila, "Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1006–1018, 2016.

[12] J. Alwen, S. Coretti, and Y. Dodis, "The double ratchet: Security notions, proofs, and modularization for the signal protocol," Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 129-158, 2019.

[13] "Signal," Signal.org, Available online at: https://signal.org/docs/ [Accessed Dec. 15, 2020].

[14] M. Marlinspike, "WhatsApp’s Signal Protocol integration is now complete,” Signal.org, Available online at: https://signal.org/blog/whatsapp-complete [Accessed Dec. 15, 2020].

[15] "WhatsApp Encryption Overview," Technical white paper, WhatsApp.com, Available online at: https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf [Accessed Dec. 15, 2020].

[16] “Facebook Messenger Deploys Signal Protocol for End-to-End Encryption,” Signal.org, Available online at: https://signal.org/blog/facebookmessenger [Accessed Dec. 15, 2020].

[17] “Open Whisper Systems Partners with Google on End-to-End Encryption for Allo,” Signal.org, Available online at: https://signal.org/blog/allo [Accessed Dec. 15, 2020].

[18] J. Lund, “Signal Partners with Microsoft to Bring End-to-End Encryption to Skype,” Signal.org, Available online at: https://signal.org/blog/skype-partnership [Accessed Dec. 15, 2020].

[19] N. Borisov, I. Goldberg, E. Brewer, “Off-the-record communication, or, why not to use PGP,” aProceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, pp. 77-84, 2004.

[20] J. Bobrysheva, S. Zapechnikov, "Post-Quantum Security of Messaging Protocols: Analysis of Double Ratcheting Algorithm," the 2020 IEEE Conference of Russian Young Researchers in

Electrical and Electronic Engineering (EIConRus), pp. 2041-2044, 2020.
[21] H. Karbasi, S. Shahpasand, "SINGLETON: A Lightweight and Secure End-to-End Encryption Protocol for the Sensor Networks in the Internet of Things based on Cryptographic Ratchets," The Journal of Supercomputing, vol. 77, no. 4, pp. 1-39, 2021.
[22] H. Karbasi, S. Shahpasand, "A Post-quantum End-to-end Encryption over Smart Contract-Based Blockchain for Defeating Man-in-the-Middle and Interception Attacks", Peer-to-Peer Networking and Applications, vol. 13, no. 5, pp. 1-19, 2020.
[23] V. Buterin, "Ethereum Whitepaper," Ethereum.org, 2014, Available online at: https://ethereum.org/en/whitepaper/ [Accessed Dec. 15, 2020].
[24] P. Zhang, F. Xiao, X. Luo, "A Framework and DataSet for Bugs in Ethereum Smart Contracts," the 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 139-150, 2020.
[25] SIDH GitHub, Available online at: https://github.com/Microsoft/PQCrypto-SIDH [Accessed Dec. 15, 2020].
[26] T. Durieux, J. F. Ferreira, R. Abreu, P. Cruz, "Empirical review of automated analysis tools on 47,587 Ethereum smart contracts", Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, pp. 530-541,2020.
[27] "Smart Contract Weakness Classification and Test Cases." SWC Registry, Available online at: https://swcregistry.io/ [Accessed Dec. 15, 2020].
[28] A. Dika, M. Nowostawski, "Security vulnerabilities in ethereum smart contracts", the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 955-962, 2018.

۱ Double-Ratchet
۲ Public Key Infrastructure
۳ Elliptic Curve Cryptography
۴ Passive Post-Quantum Attacks
۵ Off-the-record
۶ Extended Triple Diffie-Hellman
۷ Instant messaging
8 InterPlanetary File System
9 Contract-oriented
10 Content identity
۱۱ Key derivation function
۱۲ Ratchet step
۱۳ Handshake
۱۴ Deployed
۱۵ Revert
16 Getter functions
۱۷ Concolic Analysis
۱۸ Taint Analysis