Online at Will: A Novel Protocol for Mutual Authentication in Peer-to-Peer Networks for Patient-Centered Health Care Information Systems

Imrana Abdullahi Yari
Friedrich-Alexander University
imrana.abdullahi.yari@fau.de

Tobias Dehling
Karlsruhe Institute of Technology
dehling@kit.edu

Felix Kluge
Friedrich-Alexander University
felix.kluge@fau.de

Bjoern Eskofier
Friedrich-Alexander University
bjoern.eskofier@fau.de

Ali Sunyaev
Karlsruhe Institute of Technology
sunyaev@kit.edu

Abstract

Patient-centered health care information systems (PHSs) on peer-to-peer (P2P) networks promise decentralization benefits. P2P PHSs, such as decentralized personal health records or interoperable Covid-19 proximity trackers, can enhance data sovereignty and resilience to single points of failure, but the openness of P2P networks introduces new security issues. We propose a novel, simple, and secure mutual authentication protocol that supports offline access, leverages independent and stateless encryption services, and enables patients and medical professionals to establish secure connections when using P2P PHSs. Our protocol includes a virtual smart card (software-based) feature to ease integration of authentication features of emerging national health-IT infrastructures. The security evaluation shows that our protocol resists most online and offline threats while exhibiting performance comparable to traditional, albeit less secure, password-based authentication methods. Our protocol serves as foundation for the design and implementation of P2P PHSs that will make use of P2P PHSs more secure and trustworthy.

1. Introduction

Patient-centered health care information systems (PHSs) are scalable information systems that leverage information technology to support patients in managing and taking an active role in their own health [1]. PHSs are not intended to replace conventional electronic health records (EHRs); they rather complement them by providing ancillary functionality [1]. Among other features, PHSs can enable patients to control release of their data during interactions with other stakeholders [2]. A US survey revealed that 80% of 800 patients are willing to take ownership of their medical data because they currently feel sidelined in the management of their data [3]. In line with large-scale efforts to re-decentralize the internet (eg, Tim Berners-Lee's Solid project[1]), peer-to-peer (P2P) designs are also becoming of interest to PHS developers as an alternative to offline USB storage, distributed ledger technologies, or centralized databases [4]. P2P PHSs, for example, Doc.ai[2] or OnePatient,[3] promise to be less rigid and flexible and store health data locally (on any patient edge device) under the sovereignty of individual device owners. Other examples for P2P PHS are Serenity,[4] which tracks mental wellness, or decentralized systems for Bluetooth-based SARS-CoV-2 (or Covid-19) contact tracking, which ensure that users' data stays on owners' devices and notify people when they were near SARS-CoV-2 virus carriers, such as Privacy-Preserving-Proximity-Tracing in Europe [5] and Trace-Together in Singapore [6]. As envisioned by Alex Pentland et al., a paradigm shift with a focus on decentralizing information systems, such as P2P PHSs, is on the way to make information systems more resilient, flexible, and transparent [7].

On the one hand, PHSs on P2P networks improve data sovereignty because all data is technically and legally governed by patients, disrupted internet connections will not stop data access, and P2P PHSs are more resilient to single points of failure than centralized infrastructures [8,9]; Moreover, P2P network characteristics such as scalability, availability, self-configuration, and extendability suit the provision of PHSs [9]. Furthermore, P2P PHSs simplify the technical and organizational challenges to implement data regulations such as the General Data Protection Regulation (GDPR) of the

[1] https://solidproject.org/

[2] https://doc.ai/app-personalized-health-ai-companion

[3] https://refinio.net/software.html

[4] https://doc.ai/serenity-mental-wellness-companion-mood

URI: https://hdl.handle.net/10125/71079
978-0-9981331-4-0

European Union [10]. On the other hand, provision of PHSs on P2P networks poses major security issues, such as offline dictionary, sybil, impersonation, reflection, replay, parallel session, or man-in-the-middle attacks [11-14], which can impede attainment of PHS goals. P2P networks are geographically dispersed, and peers can freely join or leave the network [13]; this makes P2P networks alluring targets for attackers to wreak havoc while remaining undetected or untraceable [8]. Moreover, users need to manage information security largely by themselves [8]; a task that challenges even qualified professional administrators [8]. Additionally, the absence of a central entity to act as a trusted third party makes it more challenging to establish confidentiality and integrity [8]. From a behavioral perspective, P2P users tend to inadvertently release their sensitive information due to the complicated design of some P2P systems, inattention to appropriate resource sharing, or the set-and-forget nature of some P2P systems that run in the background [13].

Due to the high sensitivity of medical data, which is worth ten times more than credit card numbers on the black market [15], P2P PHS security issues need to be addressed to increase patient acceptance and lower the risks of using them [16]. Effective authentication protocols are highlighted as a priority among the security measures that must be employed to address P2P security issues and prevent data breaches and information loss [11,16,18]. An authentication protocol involves the use of cryptographic algorithms to validate a reported identity [12] and it assures legitimate access and authorized use of information resources [18]. In this study, we focus on the development of an authentication protocol for secure provision of P2P PHSs.

Some countries, for instance, Germany, planned to provide user authentication through smart cards for all insured persons and medical practitioners to ease security management for individual PHS providers [1,19]. However, implementation of national health-IT infrastructures is often complicated, expensive, and protracted [1]. Hence, current PHS providers should consider implementing more efficient security measures for their systems. Although the advantages of a national health-IT infrastructure for authentication could be enormous, PHS providers should be able to freely choose whether to certify their PHS based on national specifications [19]. The design of a novel authentication protocol can benefit PHS providers outside national health-IT infrastructures, for instance, in countries with less sophisticated IT infrastructures [20], and serves as a medium-term security measure for PHS providers that aim to integrate their systems with national health-IT infrastructures once they prove effective.

P2P authentication protocols that support mutual (or user-to-user) authentication exist [11,21-23]. However, state-of-the-art protocols have disadvantages that make them impractical for P2P PHSs: They rely on additional card readers [23], which can be lost; use passwords to encrypt the cards [11], which are vulnerable to offline attacks; incorporate remote cryptographic operations [24,25], which are unsuitable for P2P PHSs that should mainly run on patient devices; apply steganography for the authentication process [21], which can be detected and blocked in the network; or are dependent on biometric attributes for identification [26], which is not universally applicable because some people are, for instance, visually impaired. To address these challenges, this study proposes a novel, secure mutual authentication protocol that enables patients to have offline access and establish secure connections with other authorized parties (eg, medical practitioners or researchers) in wireless multi-hop networks while ensuring protection against offline and online threats when using P2P PHSs. Since PHSs are diverse, offered by multiple parties, and can provide any functionality patients find useful [1], our protocol design relies on a federated architecture for various PHS providers. Furthermore, to incorporate features of the proposed German national health-IT infrastructure [1] into the design, we use a software-based smart card (virtual card or vCard) feature in the authentication process. Since P2P PHSs should predominantly run on patient's edge devices, they should not require resource-intensive operations; hence, resource specifications of patient devices were a paramount consideration in the design of the protocol.

The design of the proposed secure authentication protocol for P2P PHSs was guided by formal methods for developing authentication protocols [12] and password hardening techniques [27]. The protocol has low computational cost due to symmetric encryption, has stateless and independent data and vCard encryption keys per user, and provides offline data access and mutual authentication. As proof of concept, we implemented the crucial parts of the protocol, password hardening and encryption services, using an opensource networking and cryptographic library (Libsodium) and demonstrated the feasibility of the protocol with a web application. Password hardening is used to make passwords unsusceptible to offline attacks; the independent and stateless symmetric keys are used for encrypting PHS data and the vCard. The security evaluation shows that the protocol can resist offline dictionary, man-in-the-middle, Sybil, impersonation, and typical authentication protocol attacks, such as replay message, parallel session, and reflection attacks [12,14].

2. Related research

In general, identity authentication can be done based on the following factors [12,18]: what users are (eg, behavior or biometrics), what users have (eg, smart cards or credentials), and what users know (eg, personal identification numbers (PINs) or passwords).

Traditional password-based authentication is widely applied in client-server architectures [12] and P2P systems—because it is convenient. Although Bill Gates heralded the vulnerabilities of password-based authentication already in 2004 at the RSA security symposium [18], 65% of people still used the same password for different accounts in 2018 [28] and compromised passwords were responsible for 81% of data-related breaches in 2018 [29]. Also, reused passwords contributed to 572 security incidents in 2019 in the US health sector; 41 million patient records were affected [30]. Using strong passwords and storing them in a hashed and/or an encrypted form are common methods that are applied to tackle such security issues (Figure 1); however, such methods are arguably an ineffective solution since they are susceptible to offline attacks [24,25,27].

Identity authentication protocols in wireless ad hoc and P2P networks exist. However, low entropy of used passwords makes them vulnerable to offline attacks [11] and use of steganography in the authentication process can be detected and blocked in the network [21]. Moreover, protocols that rely on public-key cryptography [22] can be computationally too expensive for P2P PHS. Recently, a user-centered identity management system was presented [26] for virtual identity derivation and biometrics, but it aims for user-host settings; therefore, it does not focus on mutual authentication, encryption services, or protection against offline attacks.

As a countermeasure to offline attacks and security issues of traditional password-based authentication protocols, password-hashing authentication leveraging remote cryptographic operations to harden the password and protect against offline attacks can be employed [24]; however, emerging services like P2P PHSs, which run locally on patient devices, have a mismatch in their underlying philosophy with remote cryptographic services such as those offered by Facebook [24]. Encryption services can be added to password hardening approaches [25], but they are unsuitable for user-to-user authentication and impractical for P2P PHS due to third-party dependencies. Moreover, some authentication protocols use anonymous credential systems designed using zero-knowledge proofs [25,26,31], which may be an undesirable feature for P2P PHS with respect to the patient-practitioner relationship during off- or online treatment where the identity of patients and other private information needs to be explicitly exposed to practitioners.

Our approach extends the idea of leveraging already available resources in a web application to protect secrets, such as passwords, against offline attacks [27]. Initially, we rely upon what users know to derive encryption services and ensure that individual users can benefit from self-sovereign authentication in accessing their data offline without any support from PHS providers. Additionally, we leverage password hardening to address the offline vulnerabilities and, in line with the P2P spirit, our protocol does not depend on any central external entity for the required computations. Moreover, acknowledging the merits of emerging nationwide health-IT infrastructures that plan to offer user authentication using smart cards, we incorporate cryptographic chunks of user identity and other control information in a vCard to ease integration of P2P PHS with future security infrastructures in the health care domain.

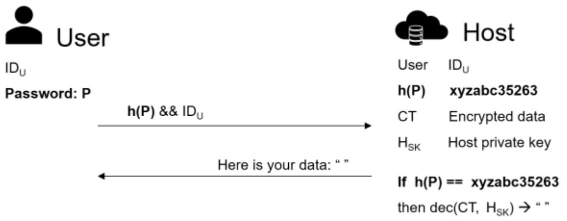

Figure 1. Typical password-based authentication. A user with a password *P* requests access to his stored data. The host stores the data in an encrypted form *CT*. When the hash (h) of *P* matches with the one stored, the host decrypts *CT* using its private key H_{SK} and returns the raw data to the user.

3. Authentication protocol development

We use Alice as a patient who owns her P2P PHS, Bob as a practitioner who wants to mutually authenticate with Alice to access her health records and Trent-1 (supporting personal health records and tracking mental health), Trent-2 (supporting pregnancy due date calculation) and Trent-3 (supporting personal health records) as independent PHS providers who host various PHSs. Trent (1:n) are federated authorities that trust each other. Arguably, national health agencies could serve as a control infrastructure to ratify various Trents in a manner similar to current practices for ratifying X.509 certification authorities [12].

3.1. Theoretical background

Key construction techniques for building identity authentication protocols are data-origin authentication, entity authentication, and authenticated key establishment [12]. Data-origin authentication aims to establish the integrity of the message using message age identification and manipulation detection in such a way that old messages may have valid data integrity but fail authenticity checks. Entity authentication is a communication process for establishing trust concerning a claimed identity. Authenticated key establishment is a process for using cryptographic keys to establish further secure communication at the application level.

In our protocol design, we used a cryptographic nonce (‘number used once’) and cryptographic hash functions for data-origin authentication. We avoided the use of timestamps because time is not necessarily synchronized between P2P nodes. For entity authentication, we rely on a trusted network of Trents to run the authentication and key establishment for their subset of users, which is a standard architecture for establishing a key agreement between Alice and Bob in wireless multi-hop networks [12]. Long-term symmetric and independent keys are used for establishing authenticated keys for communication of Trents with Alice or Bob, short-term keys are used for secure communication between Alice and Bob in a multi-hop network.

3.2. Cryptographic techniques

Assuming that $i \ \& \ j \in I$ are the respective identities of Alice and Bob, who share a secret symmetric key *K* of size $k \in \mathbb{N}$, that *N* is the nonce of Alice who is initiating the mutual authentication request, and that *conv* refers to the protocol conversation history with all encountered and authenticated users, the execution of Alice' protocol oracle $\prod(1^k, i, j, K, conv, N)$ will yield the chunk $m,\ K_{AB}\ \&\ \delta\ \in \{0,1\} * \cup \{"no\ output"\}$, where *m* is a message to be sent out, δ is a decision, which can be accept, reject, or undefined, of a principal (in this case, according to Bob's protocol) receiving the authentication request. If the protocol accepts the run, a secure short-term symmetric key K_{AB} is issued for secure communication between Alice and Bob.

Applying a generic one-way hash function on sensitive information and encrypting it may not lead to confidentiality and integrity since an attacker who compromised the database containing the sensitive information can perform offline brute force attacks on the database [12] and unveil users' sensitive information to wreak more havoc. Extant research employs sophisticated cryptographic techniques such as key-homomorphic pseudorandom functions but relies on remote cryptographic operations for password hardening [25]. Dependence on centralized third parties is undesirable for P2P PHSs. Therefore, we rely on services available locally on patient devices which are located in a separate repository within the PHS client software and use a keyed-hash message authentication code (HMAC or password hardening). Cryptographic hash functions are usually keyless when applied to a secret (or password); however, cryptographic hash functions can take a cryptographic key and concatenate it with a message *m* to be authenticated to create an HMAC. The key is not meant for encrypting *m*, rather it allows a user in possession of the correct key, the original *m*, and the hash function to compute the same output (digest) to authenticate m. Accordingly, let $h[m]^K$ represent a pair $m, prf^K\ (m)$ where $prf^K: \{0,1\} * \mapsto \{0,1\}^K$ is a pseudo-random HMAC which can be keyed with the hash of *m* for higher integrity and protection against offline attacks.

4. Results

4.1. Overview of the protocol

The overview of the protocol is shown in Figure 2. Within the remainder of the manuscript, we refer to any module that is local on a patient's device that provides strong cryptography as crypto module. PHS providers (Trents-1:n) form a federated, structured P2P network, where peers' public identities (registered Alices' and Bobs') are securely maintained under the control of distributed hash tables (DHT) [9]. The entire index is equally distributed among participating Trents; each Trent has to maintain it for lookup functionality. Network communications between Trents can be handled via an end-to-end-encrypted public-key infrastructure.

Our core scenario is that Alice wants to register with Trent-1 while Bob is registered with Trent-3. Trent-1 can issue and manage tokens that have a limited validity to Alice after they personally verify that they are eligible to use the PHS. In the case of Alice, token issuance can be offline in offline processes such as validation of health insurance. In the registration phase, Alice supplies her password and a valid token to the PHS client software from which the PHS locally HMACs the password in the crypto module and derives a stateless and independent data encryption key (DEK) along with a vCard encryption key (VEK) from the password. Next, the registration request is forwarded to Trent-1 and, after validation and verification, Trent-1 issues a vCard to Alice, which is stored in her PHS client software. Bob has to pass a similar registration process as Alice with Trent-3. In the next run (ie, login phase), Alice and Bob can access their respective PHS services offline without involving any Trent. In the mutual authentication phase, when Bob wants to access Alice's PHS in a wireless multi-hop network, he can search for Alice's public identity online using the lookup functionality from Trent-3 and then request a ticket to mutually authenticate with Alice. This communication works when Trent-1 and Trent-3 support the same PHS functionality. The ticket with limited validity, containing a short-term secure key, can be issued to Bob by Trent-3 and shared with Alice via Trent-1 after validation and verification.

The mutual authentication focuses on wireless multi-hop networks. At a single-hop network level (eg, via Wi-Fi Direct or Bluetooth), when Alice and Bob are both *physically* present in the same location (eg, in a practice or hospital) and require faster PHS access, they

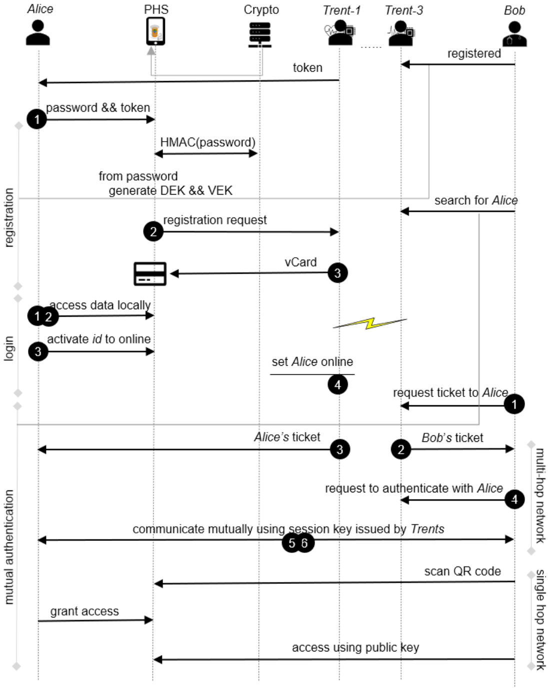

Figure 2. Authentication protocol overview

can mutually authenticate via a QR code feature on Alice's PHS client software. Bob can scan the QR code at which time his self-signed public key is shared with Alice's PHS. After establishing a network connection, Alice can permit Bob to access her health records, which allows the protocol to decrypt the records using her DEK and Bob's public key to temporarily encrypt the data. This permission can be revoked by Alice or due to network disconnection upon which the protocol removes the access rights and re-encrypts the data using Alice's DEK. This is a vital feature for P2P PHS to be independent of Trents.

4.2. Registration phase

We assume that PHSs implement a setup algorithm that activates whenever a PHS client software is first downloaded by any user, in this case Alice. Alice's self-signed private-public key pairs $[A_{SK}, A_{PK,}]$, the crypto module's private-public key pairs $[C_{SK}, C_{PK,}]$, and the public key $[T_{PK}]$ of the respective Trent (here Trent-1) are then made available to the PHS.

Initially, Alice supplies her token *to*, username *un,* and password *pw* (*flow (1): to_i || un_i || Alice || pw_i)* to the PHS client software. The private key of the PHS and the crypto module are concatenated, appended, and processed with a cryptographic hash function *h* to harden the password $h(pw_i)^{ASK \,||\, CSK}$. Random data $salt_x$ and $salt_y$ are generated and used individually with the *h(pwi)* to generate the VEK and DEK, respectively. Only the HMAC-passwords (HMAC-*pw*), $salt_{x,}$ and $salt_y$ are stored in the PHS database; $h(A_{SK})$ is stored locally within the PHS in a secure repository, the actual VEK and the DEK are stateless and only derived after each successful login. To ensure confidentiality, the VEK is used for encrypting the vCard while the DEK is used for encrypting health information and other sensitive information in the PHS. The registration request *(flow (2): $[h(to) \,||\, Alice \,||\, un_i \,||\, A_{PK} \,||\, h(VEK)]_{TPK})$,* which is encrypted using Trent's public key is forwarded to Trent-1. If Trent-1 successfully decrypts this request message using his private key while the supplied *to* is valid, Trent-1 generates a long-term secret session key K_{AT} for secure communication between him and Alice (or K_{BT} in case of Bob with Trent-3) and computes $X_A = (h(un_i \,||\, h(T_{PK})) + h(un_i \,||\, h(VEK))$ *mod 2)* and $vCard = [Alice \,||un_i \,||\, K_{AT} \,||\, X_A]_{h(VEK)}$. Trent-1 does not store the VEK of Alice. He only stores Alice's public identity which is maintained in the DHT and K_{AT}. Finally, Trent-1 sends the vCard *(flow (3): vCard,* at the provider side*)* to Alice and it is stored locally on Alice's PHS client software. From this point onwards, Alice can access her data offline without any support from Trent-1. This concept of storing private and identification information locally on owners' devices aligns well with P2P PHS goals and mitigates risks for insider threats while providing higher integrity.

4.3. Login phase

Alice's input *(flow (1): un_i || pw_i)* to log in to her PHS client software is used to retrieve the stored HMAC-*pw*. Alice's PHS uses its crypto module's and her private keys to recalculate the HMAC-*pw* with Alice's input of her password. Only if the derived *hmac-pw* digest is equal to the stored digest (HMAC-pw_i*' ==* $h(pw_i)^{ASK \,||\, CSK}$*)*, login access is granted while the VEK and the DEK are derived from the password. The DEK is used to temporarily decrypt all other stored information, the VEK is used to decrypt the vCard. In a wireless multi-hop network, Alice can activate her vCard *(flow (2): vCard,* at the patient side*)* for her P2P public identifier to be published online by Trent-1 via DHT so that other peers, such as Bob, can find her. The PHS locally computes Alice's public P2P identifier $(A_{pid} = (X_A + h(un_i \,||\, h(VEK)))$ *mod 2* $||\, N_A)$ and then forwards a request to Trent-1 who sets her P2P identifier to online *(flow (3):* $A_{pid} \,||\, Alice \,||\, h[N_A]_{KAT})$. Without depending on this request, Trent-1 computes Alice's P2P identifier using the information he already has $(A_{pid'} = h(un_i \,||\, h(T_{PK})) \,||\, N_A)$. Only when the received P2P identifier is the same as the derived one ($A_{pid'} == A_{pid,}$) will Trent-1 set Alice online. Trent then computes a confirmation response *(flow (4):*$h[N_A - 1]K_{AT})$ and sends it to Alice's

PHS client software while Alice's protocol can only accept this message if nonce N_A is valid.

4.4. Mutual authentication phase

In this phase, it is expected that Bob and Alice both logged in and activated their P2P identifiers. We assume that Bob can acquire Alice's public identifier by accessing the lookup service from Trent-3 since indexes of all participating peers are maintained in the Trents' supernode network; Bob can log the mutual authentication request *(flow (1):* $[Bob \,||\, Alice \,||\, h[N_B]]_{KBT}$*)* with Trent-3 before connecting to Alice's health record(s). After verification, Trent-3 can issue a ticket *(flow (2):* $ticket_A \,||\, [\{K_{AB}\}K_{BT} \,||\, h[N_B] \,||\, Alice \,||\, Bob \,||\, A_{pid}\,]_{KBT}$*)* to Bob while parallelly notifying Alice (via the DHT and Trent-1) with the ticket *(flow (3):* $ticket_A' = [\{K_{AB} \,||\, B_{pid}\}K_{AT} \,||\, T \,||\, Alice \,||\, Bob \,||\, A_{pid}]_{KAT}$*)* about the request. Both tickets have a secure short-term key K_{AB} which has a cryptographic association with their identities and message age identifiers N_A and N_B. If the messages are altered during transmission, changes can easily be detected. Moreover, each ticket has a valid time T so that an attacker that compromised Bob by replaying old sessions from Alice and Bob can be detected by the protocol since the old session key may have valid data integrity but fails authenticity. Next, Bob sends the ticket *(flow (4):* $ticket_A$*)* directly to Alice. She accepts this request only if the tickets received from Trent-1 and Bob are the same and T is valid. Next, Alice sends an encrypted nonce *(flow (5):* $h[N_A]_{KAB}$*)* as a challenge to Bob. To obtain N_A, Bob decrypts the challenge using K_{AB} and then sends a response *(flow (6):* $h[N_A - 1]K_{AB}$*)* to Alice. At this point, if message decryption is successful by Alice and N_A is valid, she will use K_{AB} until it expires for secure communication over a wireless multi-hop network with Bob. Finally, Alice can securely use the PHS supported by Bob and improve her treatment.

5. Evaluation

5.1. Security

For the security evaluation of our protocol, we assumed that the attacker is powerful enough and possesses all the necessary tools and techniques to eavesdrop, intercept, change, and inject malicious content in a wireless multi-hop network. Network-layer attacks require little effort. Techniques to mount such attacks leverage, for instance, WI-FI Protected Access II (WPA2) security protocol vulnerabilities [32] to conduct man-in-the-middle attacks based on free and open-source tools like Driftnet or Wireshark. The attacker focuses on the unauthorized and undetected acquisition of cryptographic credentials and not on breaking the cryptographic algorithm [12]. PHS providers should decide what cryptography to adopt (eg, AES) depending on system requirements and other factors to strengthen security. Moreover, we assume that each protocol principal (Alice, Bob, or the federated network of Trents) behaves honestly and does not expose any users' shared short-term or long-term session keys to unintended parties. Although our protocol resists many attacks, such as offline dictionary, replay, Sybil, impersonation, man-in-the-middle, parallel session, reflection, or interleaving attacks, we only present the interesting evaluations due to page restrictions. The detailed evaluation report is available from the authors upon request.

5.1.1. Offline dictionary attack

In a scenario in which an attacker compromised a PHS database containing passwords and other sensitive information, the attacker cannot successfully brute force the password even when A_{SK} is leaked since the other key used for HMACing the password (C_{SK}) is neither stored in the PHS database nor in its source code according to our protocol design and the defined assumptions. Therefore, even a password with the lowest entropy, like '12345', cannot be unveiled through offline attacks (due to HMAC). On the contrary, traditional password hashing functions like BCrypt, PBKDF2, or SCrypt [27] will merely slow down the password cracking process [27]. However, even if an attacker compromised the crypto-module's private key, our protocol HMAC uses a BLACK2 hash function, which is invulnerable to collisions, immune to length extensions, and differential enough for random oracles [33]. This serves as another layer of protection that obstructs the cracking process. Furthermore, other private information in the database, encrypted using DEK or VEK, is strongly protected since those keys are stateless and not stored anywhere—they are only vulnerable to brute force attacks. Moreover, those keys are only derived when a correct password is provided to the PHS. Consequently, attackers can only start tedious brute-force attacks on the DEK and VEK once the first hurdle has been taken.

5.1.2. Parallel session attack

This is a form of an attack in which a disgruntled insider concurrently orchestrates and executes more than one run of the protocol while blocking and replacing messages flowing from one user to the other [12]. We refer to this insider as Malice—a malicious and normal user (eg, a patient)—who also shares a secret symmetric key K_{MT} with a Trent (1:n) in the network. Even if Malice has a valid K_{MT} and initiates a simultaneous run, mutual authentication is not expected in the case of our protocol since each ticket issued by a Trent contains a secure short-term session key with a validity period that does not allow more than one user to connect to another user at the same time (this can be ensured via DHT [9]). Again, each ticket contains the identities of the authenticator and the user to be authenticated; therefore, use of a ticket issued for communication between Bob and Alice by Malice will be rejected. Although network delay can open doors for eclipse or routing attacks [14], Malice's reuse of the nonce of one user to establish a connection with another users (for example, reuse of Bob's nonce for Alice; *flow (5)* Mutual authentication phase) will fail since nonces are cryptographically and randomly generated; therefore, even if Malice can block a user (eg, through eclipse or routing attacks), a parallel session attack is impractical.

5.1.3. Impersonation attack

Assuming Malice intercepts Bob's request to be online on a P2P network and alters the message contents by replacing her identifiable information with his fake identity information *(flow (3)* in Login phase*: M_{pid} || Malice* || $h[N_B]_{KBT}$*)* with the intention of masquerading as Bob (eg, to steal patient data) and then forwarding the altered message to Trent-3 to set Bob's identity online before accepting Bob's nonce N_B, Trent-3's verification and validation of Bob's P2P identifier will fail since $B_{pid}' \neq B_{pid}$ and Trent-3 will reject the request. Again, *Malice ≠ Bob* and, consequently, identity verification fails. This method of explicitly adding identities of the participants in establishing the meaning of a message also curtails other vulnerabilities such as attacks based on name omissions [12]. Consequently, in combination with our concept of token issuance and verification, the protocol is also insusceptible to Sybil attacks.

5.2. Implementation and performance

As proof of concept, we implemented the crucial parts of the protocol, password hardening and key derivation, using a Libsodium cryptographic library. The code is available from the authors upon request. We implemented the protocol within the context of a PHS as a web application developed using PHP/MySQL. We hosted the PHS locally on an Apache web server in a virtual workstation sharing resources from a host system that uses 4-cores and 8-logical processors (AMD Ryzen 7 2700U CPU). We leveraged the existing cryptographic modssl module on Apache as crypto module to provide most cryptographic operations of our protocol. Importantly, we directly used modssl to generate C_{SK} for the password hardening. Our selection of this key is different from the secret key used for transport layer security [27]. Therefore, it is independent of any external or remote service. Moreover, we used the BLAKE2b[5] cryptographic hash function, which is faster and more secure than MD5, SHA-2, SHA-3 for the HMAC computation [33]. For key derivation, we use the award-winning password hashing function A2gon2id[6] due to its higher resistance to side-channel and time-memory trade-off attacks.

We used the Apache Benchmarking tool[7] to calculate the overhead per successful login attempt. We used two cases for evaluation. First, we used a P2P PHS with a password authentication without any hashing or key derivations. Second, we used a P2P PHS with password hardening and key derivations. For each case, we used 100 requests with 10 concurrent users and enabled HTTP keep-alive and authentication features. The first test completed in 9.337 milliseconds (ms) while the second case completed in 9.913ms; therefore, the overhead in terms of connection times per successful login attempt for using an HMAC-*pw* and key derivation on the P2P PHS is almost equal to using the default insecure authentication methods. Therefore, despite the added security provided by the proposed authentication protocol on P2P PHS, it has a similar performance with still default, albeit less secure (see evaluation steps above), authentication methods. Furthermore, we compared our proposed authentication protocol with protocols reviewed in the related research section with respect to standard authentication construction requirements, security, and other requirements for P2P PHS authentication. The summary of the comparison is shown in Table 1 and discussed in the following section. Our protocol does, not only, perform better in terms of security, but also, outperforms the existing authentication protocols in terms of their incompatibility with P2P PHSs.

6. Discussion

In the future, many health care information systems could reap the benefits of decentralization; P2P PHS are a promising, possible future development that comes with enormous advantages, such as improved privacy management, data sovereignty, and resilience to single points of failure—a new paradigm shift as pictured by Alex Pentland et al [7]. However, future P2P PHSs will introduce new challenges, such as requiring patients to manage information security for their PHSs. P2P networks pose major new security issues while inheriting other security issues that any other networked application running on the internet faces. The synergies of P2P networks and wireless multi-hop networks are

[5] https://blake2.net/

[6] https://password-hashing.net/#phc

[7] https://httpd.apache.org/docs/2.4/programs/ab.html

Table 1. Comparisons of authentication protocols with our protocol with respect to authentication requirements and other features of P2P PHS[1]

Protocol	[11]	[25]	[22]	[26]	Our
Requirement					
Data-origin authentication	+	++	++	++	++
Entity authentication	++	++	+	++	++
Authenticated key establishment	++	-			++
Protection against					
Authentication attacks[2]	+		+	+	++
Offline attacks[3]	-	++		+	+
Other					
Mutual authentication	+	-	+	-	+
Biometrics[4]	-	-	-	++	-
Anonymity		+		+	-
Encryption services	-	++	-	-	++
Virtual card feature	+	-	-	-	++
Offline data access	-	-	-	-	++
Dependent on remote crypto-module[4]	-	++	+	+	-

[1] requirement with '++' are fulfilled, '+' partially fulfilled, '-' not fulfilled, and ' ' could not be identified from the source.
[2] attacks evaluated in this study such as impersonation and replay message, parallel session, and reflection.
[3] offline dictionary attacks on passwords or sensitive information in a compromised database
[4] biometrics are problematic from a usability perspective and such identifiers cannot be replaced once compromised
[5] reliance on remote cryptographic services in the password hardening computations

known [34] but pose new security risks for P2P-PHS communications at the network level due to lack of a centralized security infrastructures and the open nature of wireless mediums [34]. Given these concerns and the high sensitivity of medical data, our study investigates the factors that could enhance the design of user authentication as a security measure for P2P PHSs. To reduce susceptibility to vulnerabilities due to abuse or offline attacks in P2P PHS, our protocol design was designed with encryption services that are stateless and independent per user. Independent encryption keys per user in a PHS also provide integrity to patients from the perspective of a PHS provider while mitigating the impacts of central attack profiles [12].

Additionally, for mutual authentication in wireless multi-hop networks, we used a federated and trusted network of PHS providers to provide authentication and key establishment in the registration phase. Subsequently, patients can access their data locally and mutually authenticate with other entities at a single-hop network level using QR codes without requiring involvement of any third party. Our protocol is interoperable since it allows for P2P PHS users of different providers while still allowing for mutual authentication; the public identities are collaboratively and securely maintained under DHT [9]. This feature can, for example, be useful for interoperability between multiple national Covid-19 proximity trackers deployed in different geographical regions. For example, when a user visits a foreign country, she can activate her Covid-19 tracker app to receive exposure notifications of diagnosed Covid-19 patients in that country while people of that country can get notifications of visitors that are diagnosed with Covid-19. Further, our protocol design is based on a decentralized approach, which ensures no entity beyond a device owner stores any personally identifiable information of the user, which addresses the privacy concerns that centralized infrastructural approaches for Covid-19 trackers are bound to cause (eg, the UK government has been criticized for wanting to store individuals' data for twenty years for their NHSX Covid-19 contact-tracking app [35]). This self-sovereign identity feature provides data protection to both the infected, non-infected, and other entities involved, increases trust, and prevents abuse of data.

Our protocol works in a federated architecture and enables mutual authentication of users of different PHS provided by different parties supporting similar PHS services. Basically, users' public information is maintained by a distributed network of supernodes (PHS providers) but under the control of DHT to facilitate issuance of authentication keys and lookup functionality. National health agencies could serve as a trusted third party to ensure and ratify that all PHS providers can be trusted and misbehaving parties are blacklisted. Such considerations are required to establish a P2P architecture suitable for PHS deployment, especially in the current situation, where countries like Germany, France, and the UK [36] identified a need to link their large health-IT infrastructures and their developing SARS-Cov-2 (or Covid-19) proximity tracking systems, although such plans may infringe user privacy [6].

Other authentication protocols for P2P systems exist [11, 21-23]; however, they either tackle isolated security concerns, are unsuitable for P2P PHSs, or do not provide independent offline data access (Table 1). In contrast to existing protocols, which assume users to be anonymous at the application and network levels, health care ecosystems can rely on established trust relationships. We avoid the use of biometric features, which are not universally applicable, in the design and focused on the use of HMACed passwords to improve usability and ensure that our authentication protocol can be adopted in P2P PHSs and used by all patients and practitioners on a global scale (Table 1). Additionally, we accounted for emerging nation-wide health-IT infrastructures [1,19]. Identifying opportunities how individual PHS providers can leverage such national infrastructures for user authentication is useful and can serve as a foundation for the design of P2P PHSs. However, such infrastructures are tedious to establish and restricted to geographic regions [1], therefore, we used a vCard (Table 1) in our protocol as a security feature in the authentication protocol design that is also useful for PHS providers outside national health-IT infrastructures and serves as a medium-term security measure for PHS providers that plan to integrate their systems with large-scale health-IT infrastructures. Such considerations serve as concepts and foundations in both theory and practice for PHS providers, PHS developers, and security service providers in the domain of PHS and P2P systems.

Our study investigates state-of-the-art solutions for mitigating security concerns of traditional authentication protocols in both P2P and user-host settings and presents useful contributions for P2P PHSs as well as the P2P network domain. Existing password hardening techniques used to address offline security issues in password authentication are unsuitable for P2P PHSs because of their dependence on remote cryptographic services (Table 1), which may affect patients' flexibility in accessing their data offline. Furthermore, state-of-the-art soluations are computationally expensive when adopted for P2P PHSs. For our protocol design, we leveraged the available cryptographic services on patients' devices to provide cryptographic operations and password hardening. We implemented the registration and login phases of the protocol but focused on password hardening and encryption services using an opensource networking and a cryptographic library (Libsodium) and demonstrated the feasibility of the protocol within the context of a PHS as a web application. We used a generic hashing function, a keyed message authentication code with a password at time of registration, and the private keys of the PHS client software and the crypto-module for the password hardening. Therefore, we supplement established theories for authentication design that provide integrity, confidentiality, and access control services with a practical utility that enables less established services like P2P PHSs to leverage already available services in such infrastructures to improve security.

Our study has limitations which offer opportunities for future research. First, our study is limited to the perspective of authentication as a security contribution for P2P PHSs. How other innovation characteristics such as usability and deployability will affect adoption by patients and health care providers is beyond the scope of this study. This is an interesting opportunity for future research investigating the behavior of P2P users and systems concerning technological maturity, which will also affect organizational decisions to adopt more secure authentication protocols. Second, the proposed protocol design has flexibility features that could be improved in many ways, for instance, by adding safety-related requirements like emergency access or guardian support. Additionally, a password change phase can be directly added to the protocol to protect against online threats such as guessing attacks. Limits for password validation requests can as well be incorporated. Reliable and patient-centered backup options to facilitate the replacement of a patient's credentials in a situation where patients lose access to their credentials (eg, a stolen laptop) can also be implemented.

7. Conclusions

P2P PHSs are an emerging phenomenon that will become more relevant in the future. So far, dedicated literature is sparse and requires research from many perspectives. With the evolving global outbreak of Covid-19, proximity tracking P2P PHSs are emerging and of growing interest for controlling the spread of the virus; however, such developments come with complications with respect to privacy risks resulting from security threats. This study takes an authentication protocol approach as a security contribution to the emerging P2P PHS landscape and is based on considerations of social aspects in health care from the perspectives of patients, practitioners, PHS providers, and large health-IT infrastructures. A global pandemic requires global solutions that go beyond national initiatives. Our protocol is interoperable and can enable users of different national implementations of proximity trackers (or other P2P PHSs) to mutually authenticate with each other over single or multi hop networks to share exposure notifications and enables health care practitioners to recommend interventions such as testing or quarantine. By being borderless, our protocol can contribute to effectively fight the Covid-19 pandemic. A secure authentication protocol could mitigate the inherent security issues of PHS deployment on P2P networks and boost the intention of patients and other stakeholders to use PHS.

We assert that our protocol is computationally secure based on the security evaluation conducted since an attacker with protocol oracle observation capabilities ($\prod_{Alice,Bob}^{s}, \prod_{Bob,Alice}^{t}\ \ s,t \in \mathbb{N}$) fails to convince a patient's (or practitioner's) protocol to accept his malicious requests due to data-origin and entity verifications [12]. Moreover, even with the added security provided by the proposed authentication protocol for P2P PHS, the evaluation shows that it has similar performance as the extant insecure authentication methods. This study can help PHS developers and providers to better understand the concepts and processes required for instantiating authentication protocols that resists most offline and online threats. Moreover, this study serves as an introduction for security service providers to the emerging landscape of P2P PHS and outlines the need for future research to curtail other prevalent security issues.

8. References

[1] Dehling, T., and A. Sunyaev. Secure Provision of Patient-Centered Health Information Technology Services in Public Networks—leveraging Security and Privacy Features Provided by the German Nationwide Health Information Technology Infrastructure, Electronic Markets, 24, (2014), 89-99.

[2] Porsche-Consulting. The Digital Revolution of the Healthcare Sector – Ecosystem, use Cases, Benefits, Challenges and Recommendations for Action. Healthcare of the Future, (2018).

[3] Spitzer, J. 63% of Americans Don'T Know Where their Medical Data is Stored: 8 Survey Insights, (2018).

[4] Sunyaev, A. Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies, Springer International Publishing, (2020), 25-49.

[5] Troncoso, C., M. Payer, J. Hubaux, M. Salathé, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, and D. Antonioli. Decentralized Privacy-Preserving Proximity Tracing, Github DP-3T Documents, 12, (2020).

[6] Cho, H., D. Ippolito, and Y. W. Yu. Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-Offs, arXiv:2003.11511, (2020).

[7] Hardjono, T., D. L. Shrier, and A. Pentland. Trusted Data: A New Framework for Identity and Data Sharing, MIT Press, (2019).

[8] Troncoso, C., M. Isaakidis, G. Danezis, and H. Halpin. Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments, PoPETs, (2017), 404-426.

[9] Vu, Q. H., M. Lupu, and B. C. Ooi. Architecture of Peer-to-Peer Systems, P2P Computing, (2010), 11-37.

[10] Gassmann, H. P. OECD Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data, Computer Networks (1976), 5, (1981), 127-141.

[11] Chen, G., H. Chen, L. Xie, G. Song, and T. Zhuang. An Identity Authentication Scheme in Wireless Peer-to-Peer Network, 12th IEE-ICCT, (2010), 473-476.

[12] Wenbo, M. Modern Cryptography: Theory and Practice, Publisher: Prentice Hall PTR, Copyright: Hewlett Packard, (2004).

[13] Liu, Z. Control Engineering and Information Systems, ICCEIS 2014, (2015).

[14] Kannengießer, N., S. Lins, T. Dehling, and A. Sunyaev. Trade-Offs between Distributed Ledger Technology Characteristics, ACM Computing Surveys, (2020).

[15] Caroline, H., and F. Jim. Your Medical Record is Worth More to Hackers than Your Credit Card, Reuters, (2014).

[16] Dehling, T., F. Gao, S. Schneider, and A. Sunyaev. Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and An-droid, JMIR mHealth and uHealth, 3, (2015). e8.

[17] Gheorghe, G., R. Lo Cigno, and A. Montresor. Security and Privacy Issues in P2P Streaming Systems: A Survey, Peer-to-Peer Networking and Applications, 4, 06/01, (2011), 75-91.

[18] Ghahramani, F., and J. Wang. Adoption of an Authentication System: Is Security the Only Consideration? ICIS 2017, (2017).

[19] Ariane, P. EHR and PHR: Digital Records in the German Healthcare System, Healthcare Industry BW, (2019).

[20] Muñoz, R. F. Using Evidence-Based Internet Interventions to Reduce Health Disparities Worldwide, Journal of Medical Internet Research, 12, (2010), e60.

[21] Xie, H., and J. Zhao. A Lightweight Identity Authentication Method by Exploiting Network Covert Channel, Peer-to-Peer Networking and Applications, 8, (2015), 1038-1047.

[22] Chen, H., L. Ge, and L. Xie. A User Authentication Scheme Based on Elliptic Curves Cryptography for Wireless Ad Hoc Networks, Sensors, 15, (2015), 17057-17075.

[23] Zhao, Z., Y. Liu, H. Li, and Y. Yang. An Efficient User-to-User Authentication Scheme in Peer-to-Peer System, IEE-INIS, (2008), 263-266.

[24] Muffet, A. Facebook: Password Hashing & Authentication, Real World Crypto, (2015).

[25] Lai, R. W., C. Egger, M. Reinert, S. S. Chow, M. Maffei, and D. Schröder. Simple Password-Hardened Encryption Services, 27th USENIX Security Symposium, (2018), 1405-1421.

[26] Bernabe, J. B., M. David, R. T. Moreno, J. P. Cordero, S. Bahloul, and A. Skarmeta. Aries: Evaluation of a Reliable and Privacy-Preserving European Identity Management Framework, Future Generation Computer Systems, 102, (2020), 409-425.

[27] Diomedous, C., and E. Athanasopoulos. Practical Password Hardening Based on TLS, Dimva 2019, International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 11543, (2019), 441-460.

[28] Google, and Harris-Poll. Google, in Partnership with Harris Poll, Surveyed a Nationally Representative Sample of 3,000 Adults (Ages 16-50+) Living in the U.S. to Understand their Beliefs and Behaviors Around Online Security. (2019).

[29] Verizon. 2018 Data Breach Investigations Report 11th Edition, (2018).

[30] Protenus. 2020 Breach Barometer. how are Health Data Breaches Affecting Your Organization? (2020).

[31] Shouqi, C., L. Wanrong, C. Liling, H. Xin, and J. Zhiyong. An Improved Authentication Protocol using Smart Cards for the Internet of Things, IEEE Access, 7, (2019), 157284-157292.

[32] Vanhoef, M., and F. Piessens. Release the Kraken: New KRACKs in the 802.11 Standard, (2018), 299-314.

[33] Aumasson, J., S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein. BLAKE2: Simpler, Smaller, Fast as MD5, International Conference on Applied Cryptography and Network Security, (2013), 119-135.

[34] Singh, M., C. Kumar, and P. Nath. Challenges and Protocols for P2P Applications in Multi-Hop Wireless Networks, 2nd ICCMC, (2018), 310-316.

[35] Alex, H. Public Health England will keep personal data of people with coronavirus for 20 years. The Guardian, (2020).

[36] Natasha, L. Germany Ditches Centralized Approach to App for COVID-19 Contacts Tracing, Techcrunch, (2020).