



(51) International Patent Classification:

H04L 9/08 (2006.01)

(21) International Application Number:

PCT/EP2021/069669

(22) International Filing Date:

14 July 2021 (14.07.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

21170182.6

23 April 2021 (23.04.2021)

EP

(71) Applicant: NEC LABORATORIES EUROPE GMBH

[DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE).

(72) Inventors: ULQINAKU, Enis; ETH Zuerich, (Eidgenös-

sische Technische Hochschule Zürich), Rämistrasse 101,

8092 Zürich (CH). **KOSTIAINEN, Kari**; ETH Zuerich, (Eidgenössische Technische Hochschule Zürich), Rämistrasse 101, 8092 Zürich (CH). **KARAME, Ghassan**; NEC Laboratories Europe GmbH, Kurfürsten-Anlage 36, 69115 Heidelberg (DE). **CAPKUN, Srdjan**; ETH Zuerich, (Eidgenössische Technische Hochschule Zürich), Rämistrasse 101, 8092 Zürich (CH).

(74) Agent: **ULLRICH & NAUMANN**; Schneidmühlstraße 21, 69115 Heidelberg (DE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO,

(54) Title: METHOD AND COMMUNICATION SYSTEM FOR SUPPORTING KEY RECOVERY FOR A USER

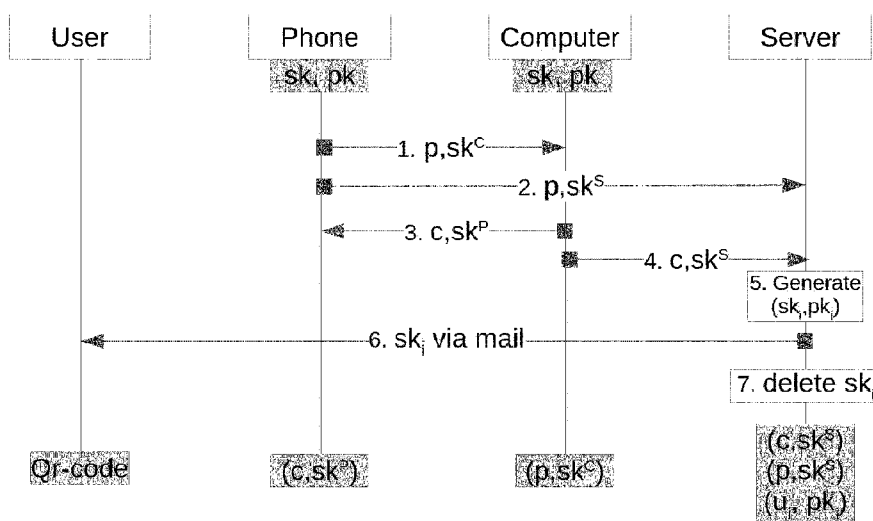


Fig. 4

(57) **Abstract:** A method for method for supporting key recovery for a user in a communication system, wherein the communication system includes an online service provider entity and at least two user devices of the user, the method comprising: performing secret sharing according to a secret sharing scheme, wherein the secret sharing scheme is configured to split a secret master key into at least two key shares that are distributed among the user devices of the user, wherein a key share of a user device of the user is split into at least two key share parts that are distributed among the online service provider entity and the other user device(s) of the user, such that a key share part is stored on the online service provider entity; generating a recovery credential for the user, wherein the recovery credential is required for retrieving the key share part from the online service provider entity. Furthermore, a corresponding communication system is disclosed.



NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW,
SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

- (84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

METHOD AND COMMUNICATION SYSTEM FOR SUPPORTING KEY RECOVERY FOR A USER

5 The present invention relates to a method for supporting key recovery for a user in a communication system.

Furthermore, the present invention relates to a communication system for supporting key recovery for a user.

10 In recent years, an increasing number of applications rely on users to safely manage secrets. For example, users of *cloud storage* need to manage secrets that can be used to encrypt files before uploading them to an untrusted cloud storage service. Users of *crypto wallets* and similar *financial applications* need to manage signing keys or similar credentials that authorize payments and other financial transactions.

15 With respect to this application, corresponding prior art documents/references are listed as follows:

20 [1] Erinn Atwater and Urs Hengartner. Shatter: Using threshold cryptography to protect single users with multiple devices. WiSec '16, 2016.

[2] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Annual International Cryptology Conference*, 1999.

25 [3] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symposium on Security and Privacy (SP)*, 2012.

30 [4] Anders Dalskov, Daniele Lain, Enis Ulqinaku, Kari Kostinen, and Srdjan Capkun. 2fe: Two-factor encryption for cloud storage, 2020.

[5] OpenJS Foundation. Node.js, accessed Mar. 2021.

- [6] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography*, 2005.
- 5 [7] Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In *ACM Conference on Computer and Communications Security (CCS)*, 2018.
- [8] Google. Advanced protection program, 2020.
- 10 [9] grempe. Github - grempe/secrets.js: Secret sharing for javascript, accessed March 2021.
- [10] J Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, 15 William Paul, Joseph A Calandrino, Ariel J Feldman, Jacob Appelbaum, and Edward W Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5), 2009.
- [11] Ponemon Institute. Airport insecurity: The case of lost & missing 20 laptops, 2008.
- [12] Ponemon Institute. The billion dollar lost laptop problem: Benchmark study of U.S. organizations, 2010.
- 25 [13] Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In *Theory and Application of Cryptology and Information Security (ASIACRYPT)*, 2014.
- 30 [14] Stanislaw Jarecki, Hugo Krawczyk, and Jason Resch. Updatable oblivious key management for storage systems. In *ACM Conference on Computer and Communications Security (CCS)*, 2019.

[15] Alasdair Mercer and Tom Zerucha. Github - neocotic/qrious: Pure javascript library for qr code generation using canvas, accessed March 2021.

[16] Microsoft. Reset or recover your microsoft account password, 2020.

[17] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[18] Victor Shoup. Practical threshold signatures. *In Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 207–220, 2000.

[19] Kenji Urushima. jsrsasign - cryptography library in javascript, accessed March 2021.

One common approach to the problem of managing secrets is to leverage user-chosen and memorable passwords. For example, the user could be asked to pick a password that is then used to derive an encryption key for cloud storage or signing key protection. However, password-based solutions have well-known serious weaknesses, including low entropy, password reuse, or memorability (cf. reference [3]). Memorability is especially critical in applications where the user might get locked out from his files or lose funds permanently, in a case where the password is forgotten.

Another common approach is to leverage full-length cryptographic keys such as encryption and signing keys generated by computers. Such secrets, obviously, provide high entropy, but storing them safely remains a significant practical challenge. In particular, storing the secret key in a single device that the user owns is risky, as the device can get lost, broken, or compromised. Simple online backup schemes will leak the secret to the online provider of the backup service. In addition, simple replication schemes, like the ones that duplicate the key to several user's devices, fail to provide key confidentiality if even one of the devices is stolen.

In view of the above, it is therefore an objective of the present invention to improve and further develop a method of the initially described type for supporting key recovery for a user in a communication system in such a way that a management of secrets, in particular secret keys, is improved within the communication system.

5

In accordance with the invention, the aforementioned objective is accomplished by a method for supporting key recovery for a user in a communication system, wherein the communication system includes an online service provider entity and at least two user devices of the user, the method comprising:

10

performing secret sharing according to a secret sharing scheme, wherein the secret sharing scheme is configured to split a secret master key into at least two key shares that are distributed among the user devices of the user, wherein a key share of a user device of the user is split into at least two key share parts that are distributed among the online service provider entity and the other user device(s) of the user, such that a key share part is stored on the online service provider entity;

15

generating a recovery credential for the user, wherein the recovery credential is required for retrieving the key share part from the online service provider entity.

20

Furthermore, the aforementioned objective is accomplished by a communication system for supporting key recovery for a user, wherein the communication system includes an online service provider entity and at least two user devices of the user,

25

wherein the communication system is configured to perform secret sharing according to a secret sharing scheme, wherein a secret master key is split into at least two key shares that are distributed among the user devices of the user, wherein a key share of a user device of the user is split into at least two key share parts that are distributed among the online service provider entity and the other user device(s) of the user, such that a key share part is stored on the online service provider entity, and

30

wherein the communication system is further configured to generate a recovery credential for the user, wherein the recovery credential is required for retrieving the key share part from the online service provider entity.

According to the invention it has first been recognized that manual identification solutions have significant cost and do not scale easily to millions of users, in

particular where an online service provider uses an out-of-band channel for identity verification of the user trying to recover a device. Therefore, it is a goal of embodiments of the invention to offer a secure recovery mechanism that reduces the frequency of manual verification operations, and thus reduces the operational cost for the online service provider.

The present invention proposes a solution that may consider the availability of a secret sharing technique, e.g., based on the sharing technique as described in reference [17]. Thus, a secret key (e.g., used for cloud storage encryption, payment authorization, or similar service) is split between two or more key shares that are distributed among the user's devices. The communication system comprises an online service provider entity and at least two user devices of the user. In this context, a secret sharing is performed according to a secret sharing scheme, wherein the secret sharing scheme is configured to split a secret master key into at least two key shares that are distributed among the user devices of the user. Furthermore, a user device again splits its key share into at least two key share parts that are distributed among the online service provider entity and the other user device of the user, such that one of the key share parts is stored on the online service provider entity. It might be provided that there are used more than two key share parts and more than two user devices. If so, the key share parts may be distributed among the online service provider entity and the other/remaining user devices of the user. Thus, each user device sends one of the generated key share parts to the online service provider entity, while the other key share part(s) is/are sent to the other user device(s) of the user.

According to the invention, a separate recovery credential is generated for the user, wherein the separate recovery credential is required for retrieving the key share part (which is stored on the online service provider entity) from the online service provider entity. For instance, in the case that a user wants to replace one of his user devices, e.g., as one of the devices is lost, the recovery credential is needed and employed when the user wants to retrieve the key share part for his new device.

Thus, the present invention provides a method and a communication system for supporting key recovery for a user, wherein the management of secrets, in particular

the management of a secret master key and its recovery, is improved. The invention and/or embodiments may provide a secure recovery mechanism that reduces the frequency of manual verification operations, and thus can reduce the operational cost for the online service provider.

5

Embodiments of the invention may provide an alternative solution to the problem of management of secrets. Thus, embodiments may provide a solution that provides better security and lower cognitive burden than traditional password-based solutions, and at the same time has better availability and theft resilience than traditional solutions that store full-length keys on user's devices.

10

The term "online service provider entity" may refer in particular in the claims, preferably in the description to a server that the user assists in a service (e.g., cloud storage and/or payments). Thus, an online service provider entity can be and/or include a server that provides online services to users. For instance, an online service provider entity may run the following services: A service may be standard *user authentication*, for example, based on username and password. The user's authentication password can be reset using standard practices such as email-based password reset. Another service may be an *identity verification*. Common examples of identity verification may include sending a copy of the ID to the service provider or showing it over a video link. Most major online service providers already have such identity verification mechanisms in place. Another service may be standard *file storage*. Thus, for instance, the online service provider entity may be a server (e.g., functioning and/or implemented as a cloud server) that provides a user authentication service, an identification verification service and/or a file storage service.

15

20

25

According to embodiments of the invention, the recovery credential may comprise a recovery key pair (sk_u, pk_u) , wherein sk_u represents a secret recovery key and pk_u represents a public recovery key. Thus, an asymmetric key pair may be provided as recovery credential. For instance, the recovery credential may be an ephemeral recovery key pair (sk_u, pk_u) .

30

According to embodiments of the invention, it may be provided that the public recovery key pk_u of the recovery credential is stored on the online service provider entity, wherein the secret recovery key sk_u of the recovery credential is stored by the user. Furthermore, it may be provided that the secret recovery key is stored by the user in a cold storage. Thus, the secret recovery key is available for the user, if a user device is lost.

According to embodiments of the invention, it may be provided that the secret recovery key sk_u of the recovery credential is encoded in a QR-code. Thus, the user can store the secret recovery key as a cold copy (e.g., on a piece of paper) independently from its user device.

According to embodiments of the invention, the recovery credential may be generated by the online service provider entity, wherein the secret recovery key sk_u of the recovery credential is delivered to the user. Afterwards, the secret recovery key sk_u is deleted on the online service provider entity. In order to provide a secure transmission to the user, the online service provider may send the secret recovery key via regular mail to the user.

According to embodiments of the invention, the online service provider entity may have a trustworthy out-of-band channel to the user, which cannot be compromised by external parties, such that the online service provider entity can securely send messages to the user. For instance, the out-of-band channel may include a regular postal delivery, such that the online service can securely send messages such as letters with sensitive information to the user. The secret recovery key of the recovery credential may be provided and stored as a cold copy such as a QR code printed on a piece of paper.

According to embodiments of the invention, the recovery credential may be generated on user-side, wherein the public recovery key pk_u of the recovery credential is delivered to the online service provider entity.

According to embodiments of the invention, the generation of the recovery credential may be performed after initialization of the user in the communication system.

According to embodiments of the invention, it may be provided that, if one of the at least two user devices is lost, the user may initiate a recovery protocol from a new user device in order to replace the lost device with the new device. This can be done while a reconstruction of the secret master key is guaranteed. The user may authenticate to the new device and may request the respective key share parts from the online service provider entity and the other user device or user devices.

According to embodiments of the invention, it may be provided that the recovery protocol includes

requesting, by the new user device, the key share part from the online service provider entity;

sending, by the online service provider entity, a challenge to the new user device;

loading, by the new user device, the secret recovery key sk_u of the recovery credential;

signing, by the new user device, the challenge with a signature based on the secret key sk_u of the recovery credential;

revealing, by the online service provider entity, the key share part to the new user device, if the signature of the challenge is valid;

receiving, by the new user device, the other key share part from the other user device of the user (or the other key share parts from the other user devices of the user).

In a scenario where a device is lost, the user cannot access the secret recovery key of the missing device, therefore he needs to reconstruct it. During enrollment one key share part of the key share is stored on the user's other device, while the other key share part is on the online service provider entity. The user may authenticate to the new user device and requests the respective key share part from both parties (the online service provider entity and the other user device). As this step might be potentially vulnerable to attackers with access to one of the user devices, the online service provider entity sends the key share part only to authenticated key share part requests. To guarantee that only the legitimate user gets the key share part, the online service provider entity may send a challenge to the new user device. The

user may scan a QR-code, which contains sk_u and signs the challenge. Finally, the online service provider entity may verify the signature with pk_u and reveals the key share part to the user only if the signature is valid.

5 According to embodiments of the invention, it may be provided that a new recovery credential is generated, if the secret recovery key sk_u of the recovery credential is lost and if the user is in control of its user devices. The previous recovery credential is thereby invalidated. Thus, the recovery enrollment can be repeated with a new recovery credential, by what the previous one is invalidated. The secret recovery
10 key sk_u may be stored in a cold storage, e.g., as a QR-code in a letter, and it is required when replacing a lost device. The loss of the secret recovery key impedes the process of device replacement in the future as the user cannot compute valid signatures. However, it may be provided that if the user is already in control of two enrolled user devices, he is able to repeat the recovery enrollment with a new key
15 pair and invalidate the old one. This process can be automated for the service provider if the user generates the new key pair (i.e. the recovery credential) locally.

According to embodiments of the invention, it may be provided that an out-of-band identity verification of the user is performed in order to guarantee that only the
20 legitimate user accesses the key share part stored on the online service provider entity, namely for the case that both the secret recovery key sk_u of the recovery credential and one of the at least two user devices is lost. The loss of the recovery secret key and one device makes difficult for the user to replace the device. Therefore, in this case the identity verification may fall back to manual verification
25 to guarantee that only the legitimate user accesses the key share parts. This process is costly for the provider because of the manual verification, however, it would be necessary only when both the recovery key sk_u and a device are lost.

According to embodiments of the invention, it may be provided that the key share
30 part is migrated between the available user device and the new user device using a local communication for the case that an available user device (i.e. one of the at least two user devices) is to be replaced with a new user device. The local communication may be provided via a local communication channel such as BLE (Bluetooth Low Energy), NFC (Near Field Communication), QR-codes, and/or

WebRTC (Web Real-Time Communication). Hence, a device replacement may be efficiently and securely performed.

5 According to embodiments of the invention, it may be provided that a proactive secret sharing of the secret master key is performed, wherein the key shares and/or key share parts are refreshed, in particular such that old key shares and/or key share parts are rendered invalid, preferably after each recovery process.

10 Embodiments of the invention may provide a new design for secure and cost-efficient key recovery without trusted third parties. The advantageous aspect of the design may include a new key recovery mechanism that combines the use of secret sharing with out-of-band authentication and cold copies (such as QR codes printed on a piece of paper). The resulting key recovery scheme in accordance with
15 embodiments of the invention may provide high security against brute force attacks, strong availability against device theft and forgotten passwords, and low operational cost for the online service provider (entity) who manages one of the recovery shares, in particular the key share part stored on the online service provider entity.

20 There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end it is to be referred to the dependent claims on the one hand and to the following explanation of further embodiments of the invention by way of example, illustrated by the figure on the other hand. In connection with the explanation of the further embodiments of the invention by the aid of the figure, generally further embodiments and further
25 developments of the teaching will be explained.

In the drawings

30 Fig. 1 is a schematic view illustrating an application scenario for a method in accordance with an embodiment of the invention, wherein the application scenario considers a system and threat model for personal encrypted cloud storage,

Fig. 2 is a sequence diagram illustrating a key derivation protocol based on Oblivious PRF, which can be run between a user's two user devices,

Fig. 3 is a schematic view illustrating a distribution of key material among user devices and the online service provider entity for a method in accordance with an embodiment of the present invention, wherein each execution of a refresh protocol causes the update of all key shares (in Layer 1 and 2) except the master key K ,

Fig. 4 is a sequence diagram illustrating an enrollment with a recovery credential that is generated on the side of the online service provider entity/server in accordance with an embodiment of the invention,

Fig. 5 is a sequence diagram illustrating exchanged messages during a recovery protocol for a method in accordance with an embodiment of the invention,

Fig. 6 is a view illustrating a user registration form for a method and a system in accordance with an embodiment of the invention,

Fig. 7 is a view illustrating a key generation for a method and a system in accordance with an embodiment of the invention, and

Fig. 8 is a view illustrating information logs during key recovery for a method and a system in accordance with an embodiment of the invention.

Before describing embodiments of the invention in detail, first, the most important aspects of a system model and assumptions that are relevant for at least some embodiments of the invention and that will probably ease their understanding are introduced. Moreover, key insufficiencies of known solutions will be outlined.

The starting point of at least some embodiments is the technique of *secret sharing* (see reference [17]). That is, the secret (used for cloud storage encryption, payment authorization, or similar service) is split between two or more shares that are

distributed among the user's devices. In practice, these two devices can be a computer/PC and a smartphone, for example. One of the shares can also be copied to an online service provider that is considered untrusted in the solution described in reference [4].

5

Embodiments may provide a new *key recovery mechanism* that combine the use of secret sharing with out-of-band authentication and "cold copies", such as QR codes printed on a piece of paper. The resulting key recovery scheme may provide high security against brute force attacks, strong availability against device theft and forgotten passwords, and low operational cost for the online service provider who manages one of the key share parts.

10

I. System Model

15

This section describes a system model that at least some embodiments of the invention may consider and some embodiments' assumptions regarding the involved entities. The exemplary system model is illustrated in Fig. 1, which shows an application scenario for a method in accordance with an embodiment of the invention, wherein the application scenario considers a system and threat model for personal encrypted cloud storage

20

User: According to embodiments of the invention, it may be considered a user who owns two devices, for example, a computer and a smartphone. It may be assumed that the user is willing to follow *simple* instruction from the online service provider entity to ensure the security and availability of his data (e.g., files stored on cloud storage), and the integrity of the transactions that can be issued using his credentials (e.g., payments issued by a digital wallet).

25

Online service provider (entity): As shown in Fig. 1, embodiments of the invention may consider an online service provider that assists the user in the service (e.g., cloud storage or payments). It may be assumed that the online service provider runs the following services: The first is standard *user authentication*, for example, based on username and password. The user's authentication password can be reset using

30

standard practices such as email-based password reset. The second service is an *identity verification*. Common examples of identity verification include sending a copy of the ID to the service provider or showing it over a video link. Most major online service providers already have such identity verification mechanisms in place (see, e.g., references [8, 16]). The third service is standard *file storage*.

Additionally, it may be assumed that the service provider has a trustworthy *out-of-band channel* to the user that cannot be compromised by the external parties. For example, this channel can be a regular postal delivery which is already established in the finance services, e.g., banking, or insurance. Using such out-of-band channel, the provider can securely send messages such as letters with sensitive information to its users.

II. Threat Model

This section describes a threat model that may be considered by at least some embodiments of the invention. The security-critical service (e.g., encrypted cloud storage or crypto wallet) faces two types of threats, as shown in Fig. 1. The first set of threats is related to the online service provider and is modeled as *cloud adversary*. The second set of threats is related to the user's devices and is modeled as *external adversary*. These two adversaries are separate entities who do not collude.

Cloud adversary: According to embodiments of the invention, it may be considered that the company running the cloud service is reputable and benign. However, the company might experience an external attack, it might have a malicious insider (such as bad administrator), or it might be coerced to release any data that it stores (e.g., request by law enforcement, subpoena). To model such threats, it may be assumed that the cloud adversary has unrestrained access to the file storage (data at rest) and he can tamper with the execution of any protocol with the user, including the user authentication and identity verification. The client software distributed by the provider to end-users is benign and can be examined by third parties.

External adversary: While most commercial solutions assume an untrusted file storage on the cloud, they typically implicitly assume that the user's devices are trusted. However, in reality users often lose their devices (cf., e.g., reference [11]), and thousands of laptops are stolen or go missing daily (cf., e.g., reference [12]) or get accessed by third parties when left unlocked (cf., e.g., reference [12]). Malware is also prevalent in commodity computing devices such as PCs and smartphones. According to embodiments of the invention, it may be considered such threats and it may be defined that the external adversary may have the following capabilities:

- *Stolen device:* The external adversary can steal the user's device, and thus access all the files on its hard drive. It is assumed access to any data at rest, because techniques such as full-disk encryption are not yet in prevalent use, especially outside the corporate domain. Full-disk encryption is an option feature in popular consumer PC platforms like Windows 10. Further, some Windows instances enable password-less full-disk encryption only, and thus the attacker would only need to turn on the device to get the disk decrypted. Because of the high risk associated with the loss of the disk password, it is unlikely that password-protected full-disk encryption will be enabled by default for all users.
- *Temporary access:* The external adversary may also gain temporary access to one of the user's devices that was left unlocked or circumvent the used screen lock protection (see, e.g., reference [12]) – also known as “evil maid attacks” (see e.g., reference [2]). With temporary access, the external adversary can use the device on behalf of the user for a short amount of time (more advanced scenarios, such as cold-boot attacks as described in reference [10] to extract all RAM content, are out of scope).
- *Malware infection:* The external adversary may also be able to infect one of user's devices with malware. This is the most powerful scenario, where the external adversary has access to all files stored on the hard drive and all data in memory (such as open session tokens), and he learns any user input that is provided to the device (e.g., user's password).

According to embodiments of the invention, it may be assumed that an adversary can compromise *at most one* of the user's devices at a time. It is noted that in these cases, the attacker's aim may be to access data that is not already stored in the device in plaintext.

5

III. Baseline Solution: Secret Sharing

This section outlines a simple secret sharing scheme that may be used as a baseline for at least some embodiments of the invention. On the one hand, password-based key derivation schemes do not provide strong confidentiality due to low entropy and forgotten passwords may render files or financial assets permanently inaccessible. On the other hand, approaches that use full-length keys and copy them into multiple devices improve usability and availability when a device gets lost, but they harm security – an adversary can compromise any of the devices to learn the key. Therefore, it is looked into alternative solutions and *secret sharing* (see, e.g., reference [17]) may be taken as the basis of a solution in accordance with embodiments of the invention.

10

15

20

25

Existing solutions such as 2FE as described in reference [4] and Shatter (see reference [1]), generate full-length pseudorandom keys on the users devices and distribute the secrets among the two user's devices. Besides secret sharing, such solutions use *oblivious pseudo-random functions* (PRFs) and *threshold signatures* (see references [14], [13], [18], [7]) to reconstruct the key or to perform distributed operations on it when needed.

30

An overview of an example scheme based on secret sharing and oblivious PRF is shown by Fig. 2, which is a sequence diagram illustrating a key derivation protocol based on Oblivious PRF, which can be run between a user's two user devices. The sequence diagram of Fig. 2 represents the exchanged messages during the key derivation protocol.

It is assumed that both the user's smartphone and the computer have one key share of the master secret K . An example use of such scheme is to encrypt a message

for cloud storage. Step 1 of Fig. 2 illustrates this, where the user inserts a message to the computer. In Step 2, the two devices execute an oblivious PRF protocol (see Section III.2 for protocol details) that derives a fresh encryption key k for the inserted message. After the fresh encryption key k is derived in Step 3, it can be used to encrypt a message, which is sent to the server (as the online service provider entity) in Step 4. After that, the computer deletes k so that future theft or compromise will not be sufficient to decrypt the message.

One example solution, Shatter as described in reference [1], uses threshold signatures in a straight-forward way, it assumes the user has n devices and t of them should be present to compute a valid signature. Recovery of a device is possible as long as the user controls at least t devices. Shatter of reference [1] does not put any share on the server, therefore the system can be initialized if the user enrolls at least *three* devices (minimal conditions to perform a 2-out-of-3 threshold signature). In accordance with embodiments of the invention, this is considered inconvenient for most practical deployments and incompatible to our system model outlined in Section I.

To consider another example solution, 2FE as described by reference [4] also uses secret sharing schemes as its main building blocks. The 2FE scheme assumes that the user has two devices and uses the server as an additional “virtual” device. It uses 2-out-of-2 secret sharing scheme and splits the master secret K into two shares, K_C and K_P , that are stored on the computer and the phone, respectively. Since the server is considered untrusted for confidentiality, it does not participate in the generation of the key material for the master key K .

Fig. 3 shows a distribution of the key material among user devices and the server, i.e. the online service provider entity, for a method in accordance with an embodiment of the present invention, wherein each execution of a refresh protocol causes the update of all key shares (in Layer 1 and 2) except the master key K . It is noted that none of the devices has access to the master key K at any point. To prevent against attackers that compromise one device at one point, and another one in the future, the system uses proactive secret sharing schemes. That is, the devices perform a *refresh protocol* periodically, which renders old shares invalid. As

shown in Fig. 3, secret shares are organized in two layers. Layer 1 stores the shares of secret keys of user devices (each device acts as a dealer for its key K_C or K_P) and the *refresh protocol* is trivial (see Section III.1). In Layer 2 none of the devices has access to the master key K , however, each device should update the local keys, from K_C and K_P to K'_C and K'_P , while maintaining the invariance to the master key K . Note that, in this case none of the devices (neither the server) can act as a dealer. The secret sharing scheme for Layer 2 is described in Section III.2.

In the following, more information about cryptographic building blocks and tools that are used in the baseline solution for embodiments of the invention is provided.

III.1 Shamir Secret Sharing

Unfortunately, simple 2-out-of-2 secret sharing has the following data availability problem. If one of the user's devices is lost or stolen by the external adversary, reconstruction of the full encryption secret K becomes impossible, and the user will permanently lose access to all his data. To address this problem, 2FE (see reference [4]) uses the untrusted cloud storage as a *virtual* third device. After the master key is generated, both user's devices split their share into two parts and send one share to the other user's device, while the other share to the server. For instance, the phone computes a 2-out-of-2 with Shamir's secret sharing for its key K_P and sends one share K_P^C to the user's computer, while the other share K_P^S to the server (the user's computer does the same). Therefore, if the user's phone is lost or stolen, the user can recover the needed key material from the cloud storage and the other device.

In order to recover a lost device, 2FE (see reference [4]) requires an out-of-band identity verification of the user to the remote server, otherwise the server does not reveal the key share. This step prevents an adversary that controls one of the user's devices to trigger the recovery process. However, the out-of-band authentication is usually manual and therefore costly for service providers.

III.2 The Oblivious Pseudo-Random Function

Secure key derivation: The main goal of schemes that distribute the secret keys among user's devices is to protect the master key in cases where one of the devices is compromised by an external adversary. For this, a per-file encryption key k should be derived as $k = F(K, s)$ where F is a pseudo-random function (PRF) and s some fresh randomness that ensures two files get different keys. The key derivation process should be such that K_C is not revealed to the phone (or K_P to the computer). Thus, K_P cannot simply be sent to the computer in order to compute F on its own as that would reveal K .

To address this challenge, reference [4] computes F in an "oblivious" fashion. An *Oblivious* PRF (OPRF) (cf. reference [6]) is a protocol where one party holds a key K while the other holds an input x . The output is the value $z = F(K, x)$ which is given to the party who holds x . However, reference [4] slightly modifies the oblivious protocol that goes as follows: The computer has input (K_C, s) while the phone has input (K_P, s) . The output $k = F(K_C + K_P, s) = F(K, s)$ is given to the primary device, without revealing anything about K_C to the secondary device (or K_P to the primary device). This protocol is described below in Protocol 1.

Protocol 1: Oblivious Key Derivation

1. Let $x \in \{0,1\}^*$ be the input, given to C and P .
2. P computes $B = K_P \cdot (H'(x))$. Send B to C .
3. When C receives the value B , C computes and outputs key k as $k = H(x, B + K_C \cdot (H'(x)))$.

Proactive secret sharing of the master key K : It is now started by describing the specifics of the secret sharing of K that is compatible with the instantiation of the Protocol 1 and that supports key share updates used in reference [4]. The primary and secondary each generate at random a key share K_C and K_P , both elements of the finite field $\mathbb{Z}_p [0, \dots, p-1]$. Note that this defines the master encryption key $K = K_P + K_C$ where addition is modulo p . It is noted that each device secret-shares their local key using Shamir secret sharing as described in Section III.1.

An attacker that compromises user's devices during different times of operation would eventually learn both shares of K . Reference [4] describes to perform key share refresh after each recovery process, which invalidates the old key shares.

5 Therefore, an adversary that compromises one user device, leaks the shares stored on it, and later compromises the other device cannot compute the master key K if refresh happened between the two events. Refreshing keys is performed by first adding a secret-sharing of 0 to each key share, using the aforementioned property that additions can be computed over shares. Say, C picks a random value z and

10 then updates its key to be $K'_C = K_C + z$ and sends $z' = -z$ to P who sets $K'_P = K_P + z'$. Note that $(K_C + z) + (K_P - z) = K + 0 = K$, so the invariant is maintained. However, if the adversary steals K_C and then $K'_P = K_P - z$, he will not learn anything about K as z is unknown.

15 IV. Secure Key Recovery in accordance with embodiments of the invention

In this section, a new scheme for secure key recovery is presented for scenarios where one key share is stored in the server. It is assumed a Baseline solution similar

20 to 2FE of reference [4], as described in Section III., where the online service provider uses an out-of-band channel for identity verification of the user trying to recover a device. However, manual identification solutions have significant cost and do not scale easily to millions of users. Therefore, the goal of an embodiment of the invention is to offer a secure recovery mechanism that reduces the frequency of

25 manual verification operations, and thus reduces the operational cost for the online service provider.

According to an embodiment of the invention, a system model assumes that the online service provider follows the protocols and stores correctly the data submitted

30 by users, including key shares. A method and a communication system in accordance with an embodiment of the invention introduces a new credential that is delivered to the user after initialization and is required only when the user retrieves a key share (or key share part) from the server. Namely, during enrollment a new key pair – referred as recovery credentials – is generated. The server, i.e. the online

service provider entity, stores the public part of the key, while the user encodes the secret key (which may be also designated as private key) in a QR-code and stores it in a cold storage.

- 5 It is stressed that the recovery credentials stored as QR-codes do not contain any information about the user's secret keys used for encryption or signing. Therefore, the user should safely store the letter for future use in case a device goes missing, but it is not critical for the security of the user's account. This solution scales easily to a large number of users and avoids triggering the manual identity verification
- 10 always when the users loses one of his devices. The following table summarizes the different authentication primitives the online service provider has available and outlines the operational cost of each primitive:

Identity Service	Cost	Occurance	Functionality
Passwords	low	often	authentication
QR-Code	low	occasionally	device recovery
Manual	high	very rarely	QR-code and device recovery

- 15 In an embodiment according to the invention, the recovery credential that may be encoded as QR code and may be saved on a piece of paper resembles the common use of "cold copies" in cryptocurrency wallets. For example, many cryptocurrency wallet services recommend that their users print out their key and save the paper for possible future use. Essentially, in such solutions the piece of paper contains the
- 20 master secret itself. The drawback of such scheme is that if the paper is lost, the master secret becomes permanently irrecoverable. The crucial difference in an embodiment according to the invention is that the paper with the QR code does not contain the master secret (or a key share). Instead the paper encodes a separate recovery credential. The main benefit of such different design is that in a rare but
- 25 still possible scenario where the paper is lost permanently, the master secret is not lost permanently, as it can still be recovered through (more expensive) manual out-of-band identity verification process. That is, the permanent loss of the recovery credential does not make the master secret irrecoverable which increases the robustness of the system.

IV.1 Recovery Initialization

5 In this section, two approaches for initializing a secure recovery mechanism are presented in accordance with embodiments of the invention. The initialization phase include the generation of a *recovery credential* (i.e., an asymmetric key pair) using two alternative ways depending on the user's preference.

10 **Approach 1: Server-side generation.** During user enrollment, the server (i.e. the online service provider entity) generates an ephemeral key pair (sk_u, pk_u) , which is needed only when the user replaces one of his devices. The server sends the secret key sk_u to the user via regular mail and stores only the public key pk_u . The secret key is encoded as a QR-code and printed in a letter delivered to the user who is instructed to store it safely. Afterwards, the server deletes the secret key sk_u . It is
15 noted that the online service provider is trusted to complete this step correctly. The process may follow the following steps (see Fig. 4):

- Phone uses 2-out-of-2 secret sharing for its key (Step 1 and 2).
- Similarly, the computer secretly shares its key (Step 3 and 4).
- 20 - The server generates *recovery credentials* for the user (Step 5).
- Provider/server sends the secret key sk_i to the user (Step 6), and deletes it afterwards (Step 7).

25 Thus, the sequence diagram of Fig. 4 shows an enrollment with a recovery credential that is generated on the side of the online service provider entity/server in accordance with an embodiment of the invention.

30 **Approach 2: User-side generation.** The approach of generating the recovery key-pair on the server requires to trust the provider that it behaves honestly and sends the QR-code to the user securely. An alternative that reduces the trust on the server is to generate the key pair (sk_u, pk_u) on the user side and send only the public part pk_u to the server. The user is liable to store safely the secret key sk_u , e.g., encode as a QR-code and print it in a letter for future use. (This approach could integrate well with FIDO protocols also.) This way, the service provider does not have access

to the sk_u at any point. However, the server still has access to the key shares and can reveal them to any colluding party. Section V. details an upgrade that removes the trust from the provider by leveraging server side TEEs.

5 IV.2 Key Recovery

10 A communication system in accordance with an embodiment of the invention allows users to replace a device while guaranteeing the reconstruction of the master key. The communication system may support device replacements as described in the following scenarios:

15 **Device is lost.** In this scenario, the user cannot access the secret key share of the missing device, therefore he needs to reconstruct it. As described above with regard to an embodiment of the invention, during enrollment one key share part of the secret key share is stored on the user's other device, while the other key share part is on the server. The user may authenticate to the new device and may request the respective key share part from both parties (the server and the other user device).

20 As this step might be potentially vulnerable to attackers with access to one of the user devices, the server sends the key share part only to authenticated key share requests (see Section IV.1). To guarantee that only the legitimate user gets the key share, the server sends a challenge to the new device. The user should scan the QR-code, which contains the secret recovery key sk_u and signs the challenge. Finally, the server verifies the signature with the public recovery key pk_u and reveals the key share part of the key share only if the signature is valid.

25 The protocol may comprise the following steps (see Fig. 5):

- 30 - The user initiates the recovery protocol from the new device (Step 1).
- The server sends a challenge to the new device (Step 2).
- The user loads the *recovery credentials* and signs the challenge (Steps 3 and 4).
- If the signature is valid, the server reveals the secret key share part sk^S (Step 5).

- The new device receives the other key share part sk^C from the other user's device (Steps 6 and 7). Finally, the new device can construct the key share from the two key share parts.

5 Thus, the sequence diagram of Fig. 5 shows exchanged messages during a recovery protocol for a method in accordance with an embodiment of the invention. Specifically, Fig. 5 shows the recovery protocol for the phone. A similar procedure may be followed when recovering the computer.

10 **QR code is lost.** The secret recovery key sk_u is stored in a cold storage, e.g., as a QR-code in a letter, and it is required when replacing a device. The loss of this key impedes the process of device replacement in the future as the user cannot compute valid signatures. However, if the user is already in control of two enrolled devices, he is able to repeat the recovery enrollment with a new key pair (see Section 5.2)
15 and invalidate the old one. This process is automated for the service provider if the user generates the new key pair locally.

Device and QR are lost. The loss of the recovery secret key and one device makes difficult for the user to replace the device. In this case the identity verification falls
20 back to manual verification to guarantee that only the legitimate user accesses the key share parts. This process is costly for the provider because of the manual verification, however, it is necessary only when both the secret recovery key sk_u and a device are lost.

25 **Device replacement (old device available).** This scenario assumes the user has access to the old device that should be replaced and to the new one that is to be enrolled. This case is simple as the user can migrate the secret (e.g., the key share) using local communication between the two devices. Options for the communication
 channel might be: BLE, NFC, QR-codes, or WebRTC. After key migrations, devices
30 perform a key refresh protocol to invalidate old key shares.

V. Leverage server-side TEE

In this Section extend the recovery mechanism is extended for server-side trusted execution environment (TEE) support for additional security guarantees. The generation of the recovery key pair on the user side simplifies significantly the process previously implemented by the server (i.e., the online service provider entity). Now, the server needs only to store a public recovery key for each user and verify a signature for every key share request. This approach allows to fully implement the server side component in TEEs and guarantee that only the legitimate users that own *recovery credentials* can access their key shares. A secure TEE rejects key share requests from unauthorized parties, including malicious actors inside the service provider itself, therefore reducing further the trust to providers for managing the key shares properly.

One challenge that rises with this approach is when the user loses access to one of his devices and the QR-code (*recovery credentials*) simultaneously, the fallback procedure is the manual identity verification. But, if the TEE only accepts valid signatures to reveal the key shares, the user cannot complete the key recovery procedure. To avoid this scenario, the TEE program includes the public key of the core entity inside the provider that deals with the manual identity verification. To avoid any illegitimate access of key shares by the provider, the TEE program logs every request, i.e., to a third-party auditor or a blockchain, before revealing the key shares. The log includes information if the recovery is triggered automatically by the user or through the manual verification mechanism, therefore a user could parse the logs related to his account and detect any malicious behavior if they exist. In the following Listing a skeleton of an exemplary TEE program is provided:

```
1  users_pk = [pk_u1, pk_u2, pk_u3, ..., pk_uN]
2  manual_recovery_pk = pk_0
3
4  //u_id (user id), d_id (device_id), ksh (key share)
5  bool function storeShares (int u_id, int d_id, int* ks) {
6      ...
7  }
```

- 25 -

```

8
9  //store user's public key of recovery credentials
10 bool function storePK (int u_id, int* pk) {
11     ...
5  12 }
13
14 //s (signature)
15 *int function deliverShare (int u_id, d_id, int* s ) {
16     if (!valid(s))
10  17         return null;
18     ...
19     return ks;
20 }

```

15

VI. Prototype Implementation

In this section, a prototype implementation of the system presented in Section IV. is described. The prototype in accordance with an embodiment of the invention has

two main components: one runs on the server side and manages user authentication and secret share delivery; while the other runs on the user side and manages local keys of user devices, computes the respective key shares, and combines shares to recompute a key during recovery.

Server side. The server is implemented in *NodeJS v10.19* with the *express* framework (see reference [5]). The server makes use of web sockets (*socket.io* package) to simplify the message passing between the parties. Further, the *crypto* module of *NodeJS* is employed for the cryptographic functionalities required in the server, i.e., the *verify* function to validate the signature submitted by a device during recovery. Finally, the following packages are used: *bcrypt*, *body-parser* and *cookie-parser*, for additional functionalities. The prototype implements a minimal user database as a JavaScript array to simplify the installation, and prints it in *db.json* file to prevent data loss if the server restarts.

Client side. This component is implemented as a web application using JavaScript and HTML, therefore it executes on any operating system running a browser. The device keys are generated locally and stored via Web Storage APIs. Further, third-party libraries are used to perform the cryptographic functionalities required by the system. *Secrets.js* (see reference [9]) provides the following functions: *key = secrets.random(length)* to generate a key locally; *shares = secrets.share(key, n, t)* computes *n* shares of *key* with a threshold *t* based on Shamir's secret sharing; and *recovered_key = secrets.combine(shares)* which recomputes the *key* given at least *t* shares.

The prototype generates the recovery credentials locally during user registration, which refers to the second approach presented in Section IV.1. The *jsrsasign* library (see reference [19]) in JavaScript is used to perform the required cryptographic functionalities. The recovery credentials consist on a key pair that is computed with the following function: *ecKeypair = KEYU-TIL.generateKeypair("EC", "P-256")*. Afterwards the private key is encoded as a QR-code using *qrious* library (see reference [15]), while the public key is sent to the server for signature verification in the future. The signature during key recovery is computed with the code in the following Listing, namely a JavaScript function to compute signatures on the user side:

```
alg = "SHA256withECDSA"; crv = "P-256";  
sig = new KJUR.crypto.Signature("alg": alg, 'curve': crv);  
sig.init(localStorage.getItem('recovery credential'));  
sig.updateString(userId + username + recoveryCnt);  
var hSigVal = sig.sign();
```

For the prototype, it is opted for third-party libraries which simplify the development and installation, however, it is stressed that a real deployment should prioritize secure APIs such as Web Crypto API when possible, which run natively and even improve the performance. Finally, the prototype works with any pair of devices, however, it is assumed the user has a computer and a smartphone, so the server can differentiate which device requests recovery and releases the respective share.

For instance, the *isMobile()* function serves to differentiate between the user's devices.

VI.1 High level functionalities

5

Registration. End users interact with the system via the client side component running locally on the browser. Users enroll in the system by completing a registration form as usual, while the JavaScript program generates a key pair in the background and displays a QR-code that encodes a private key. The public part of the key pair is appended to the form data and is submitted to the server. A screenshot of the registration form is shown in Fig. 6. The user is instructed to store the QR-code safely, e.g., storing digitally on a secure drive or printing in a paper. On a real deployment the key pair can be generated on the server side and send via post to the user (see Section IV.1).

15

Device keys generation. Once the user logs in the system by filling the respective form, the prototype shows the home page. On the first run, before the device keys are generated, the interface allows the user to generate new keys. For this step, the user should login from the both devices (computer and phone) since they need to sent the respective key shares to each other. The prototype provides logs of main events to the user as shown in Fig. 7, however, one can read more detailed logs by opening the browser *console*.

Recovery. The client interface allows the user to delete local key material, including device keys and all key shares. At this state, the device does not have access to any key material and simulates a new device which needs to recover old keys. Before initiating the recovery process the user should load the private key of the *recovery credentials* into his device, i.e., scan the QR-code with a camera app. (Most default camera apps on recent smartphones detect automatically QR-codes and redirect users to the encoded URL.) The user can trigger the recovery process by clicking the button "Recover keys of the old Device". Again, logs are presented to the user which show the received shares and the computed key (see Figure 8). The recovered key in this step should be the same as the generated key in the previous step (see Fig. 7). Note that, before repeating the recovery procedure, the

30

user should log out and enter the URL encoded in the QR-code in the address bar (or scan the code). After recovery, devices should run a *refresh protocol* to compute new shares and invalidate the old ones, however, the prototype currently does not provide this feature.

5

VII. Conclusion

The security of an increasing number of applications depend on the users' chosen secrets, especially crypto-wallets and encrypted cloud storage. While passwords are proven to lack the security guarantees required, high entropy secrets pose a critical trade-off between security and availability for users. Previous works as described in references [4] and [1] present practical systems for key distribution among user's devices to avoid having a single point of failure. However, their key recovery of a missing device becomes costly because the server should perform manual identity verification during recovery. An embodiment of the invention provides a method that automates the recovery process, therefore reducing the operating costs for the provider. For instance, this may be possible because the user and the server generate *recovery credentials* during enrollment and the user stores the secret recovery code (e.g., a QR-code) in a cold storage. The QR-code itself is not critical because it does not store information about user's master keys. Finally, an embodiment of the invention provides an update of the system that allows implementing the server side component in TEEs, therefore reducing even further the trust to service providers.

10

15

20

25

30

35

Many modifications and other embodiments of the invention set forth herein will come to mind to the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. A method for supporting key recovery for a user in a communication system, wherein the communication system includes an online service provider entity and at least two user devices of the user, the method comprising:
- 5 performing secret sharing according to a secret sharing scheme, wherein the secret sharing scheme is configured to split a secret master key into at least two key shares that are distributed among the user devices of the user, wherein a key share of a user device of the user is split into at least two key share parts that are distributed among the online service provider entity and the other user device(s) of the user, such that a key share part is stored on the online service provider entity;
- 10 generating a recovery credential for the user, wherein the recovery credential is required for retrieving the key share part from the online service provider entity.
- 15 2. The method according to claim 1, wherein the recovery credential comprises a recovery key pair (sk_u, pk_u) , wherein sk_u represents a secret recovery key and pk_u represents a public recovery key.
- 20 3. The method according to claim 2, wherein the public recovery key pk_u of the recovery credential is stored on the online service provider entity, and wherein the secret recovery key sk_u of the recovery credential is stored by the user.
- 25 4. The method according to claim 2 or 3, wherein the secret recovery key sk_u of the recovery credential is encoded in a QR-code.
- 30 5. The method according to any one of claims 2 to 4, wherein the recovery credential is generated by the online service provider entity, wherein the secret recovery key sk_u of the recovery credential is delivered to the user, and wherein the secret recovery key sk_u is deleted on the online service provider entity afterwards.
6. The method according to any one of claims 2 to 5, wherein the online service provider entity provides a trustworthy out-of-band channel to the user, such that the online service provider entity is able to securely send messages to the user.

7. The method according to any one of claims 2 to 6, wherein the recovery credential is generated on user-side, wherein the public recovery key pk_u of the recovery credential is delivered to the online service provider entity.

5 8. The method according to any one of claims 2 to 7, wherein the generation of the recovery credential is performed after initialization of the user in the communication system.

9. The method according to any one of claims 1 to 8, further comprising:

10 if one of said at least two user devices is lost, initiating, by the user, a recovery protocol from a new user device in order to replace the lost device with the new device, in particular while guaranteeing a reconstruction of the secret master key, wherein the user authenticates to the new device and requests the respective key share parts from the online service provider entity and the other user device(s).

15 10. The method according to any one of claims 1 to 9, wherein the recovery protocol includes

requesting, by the new user device, the key share part from the online service provider entity;

20 sending, by the online service provider entity, a challenge to the new user device;

loading, by the new user device, the secret recovery key sk_u of the recovery credential;

25 signing, by the new user device, the challenge with a signature based on the secret key sk_u of the recovery credential;

revealing, by the online service provider entity, the key share part to the new user device, if the signature of the challenge is valid;

receiving, by the new user device, the other key share part(s) from the other user device(s) of the user.

30 11. The method according to any one of claims 1 to 10, further comprising:

if the secret recovery key sk_u of the recovery credential is lost and if the user is in control of its user devices, generating a new recovery credential, wherein the previous one is invalidated.

12. The method according to any one of claims 1 to 11, further comprising:

if both the secret recovery key sk_u of the recovery credential and one of said at least two user devices is lost, performing an out-of-band identity verification of the user to guarantee that only the legitimate user accesses the key share part stored on the online service provider entity.

13. The method according to any one of claims 1 to 12, further comprising:

if an available user device is to be replaced with a new user device, migrating the key share part using a local communication between said available user device and said new user device.

14. The method according to any one of the claims 1 to 13, wherein a proactive secret sharing of the secret master key is performed, wherein the key shares and/or key share parts are refreshed, in particular such that old key shares and/or key share parts are rendered invalid, preferably after each recovery process.

15. A communication system for supporting key recovery for a user, in particular for executing a method according to any one of claims 1 to 14, wherein the communication system includes an online service provider entity and at least two user devices of the user,

wherein the communication system is configured to perform secret sharing according to a secret sharing scheme, wherein a secret master key is split into at least two key shares that are distributed among the user devices of the user, wherein a key share of a user device of the user is split into at least two key share parts that are distributed among the online service provider entity and the other user device(s) of the user, such that a key share part is stored on the online service provider entity, and

wherein the communication system is further configured to generate a recovery credential for the user, wherein the recovery credential is required for retrieving the key share part from the online service provider entity.

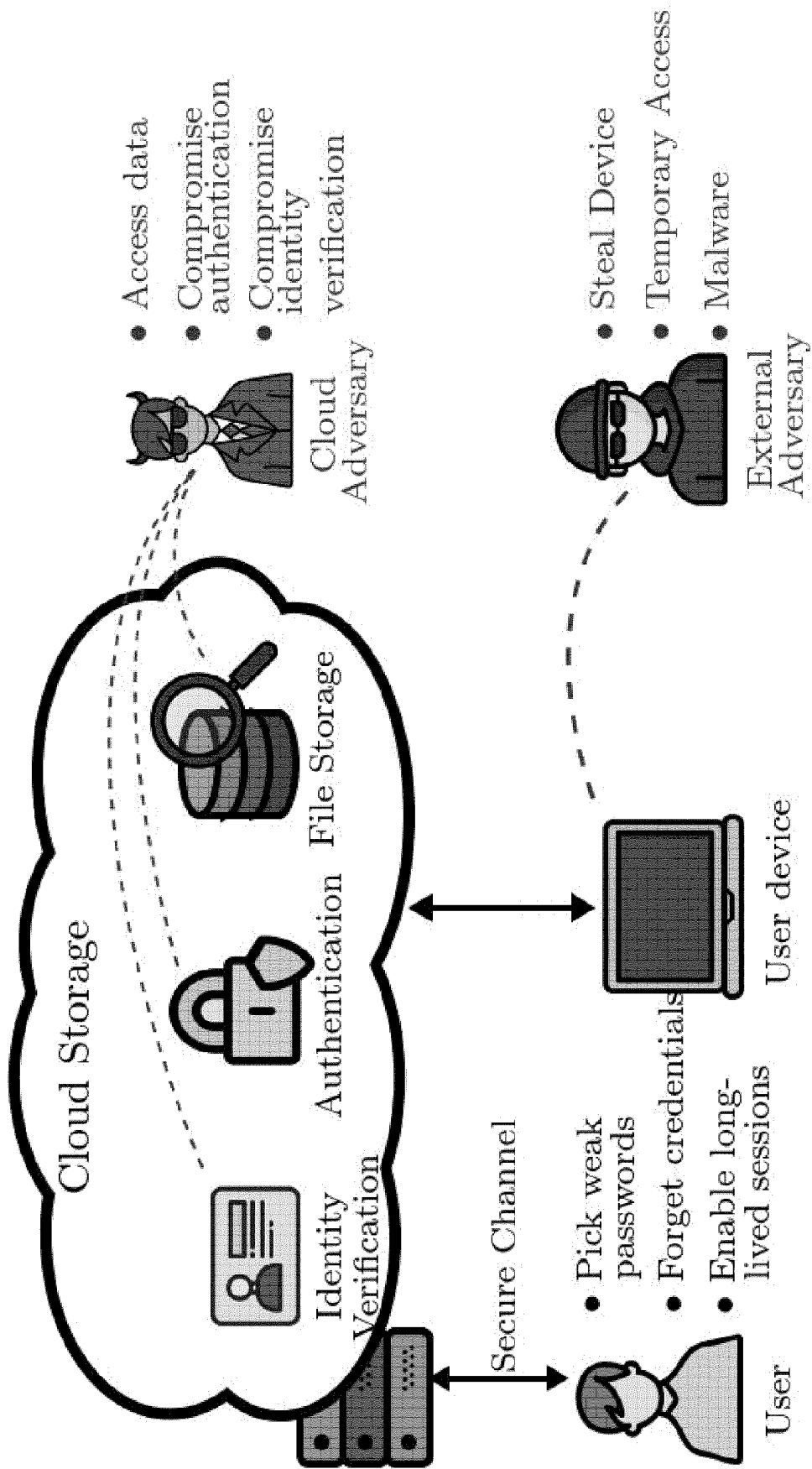


Fig. 1

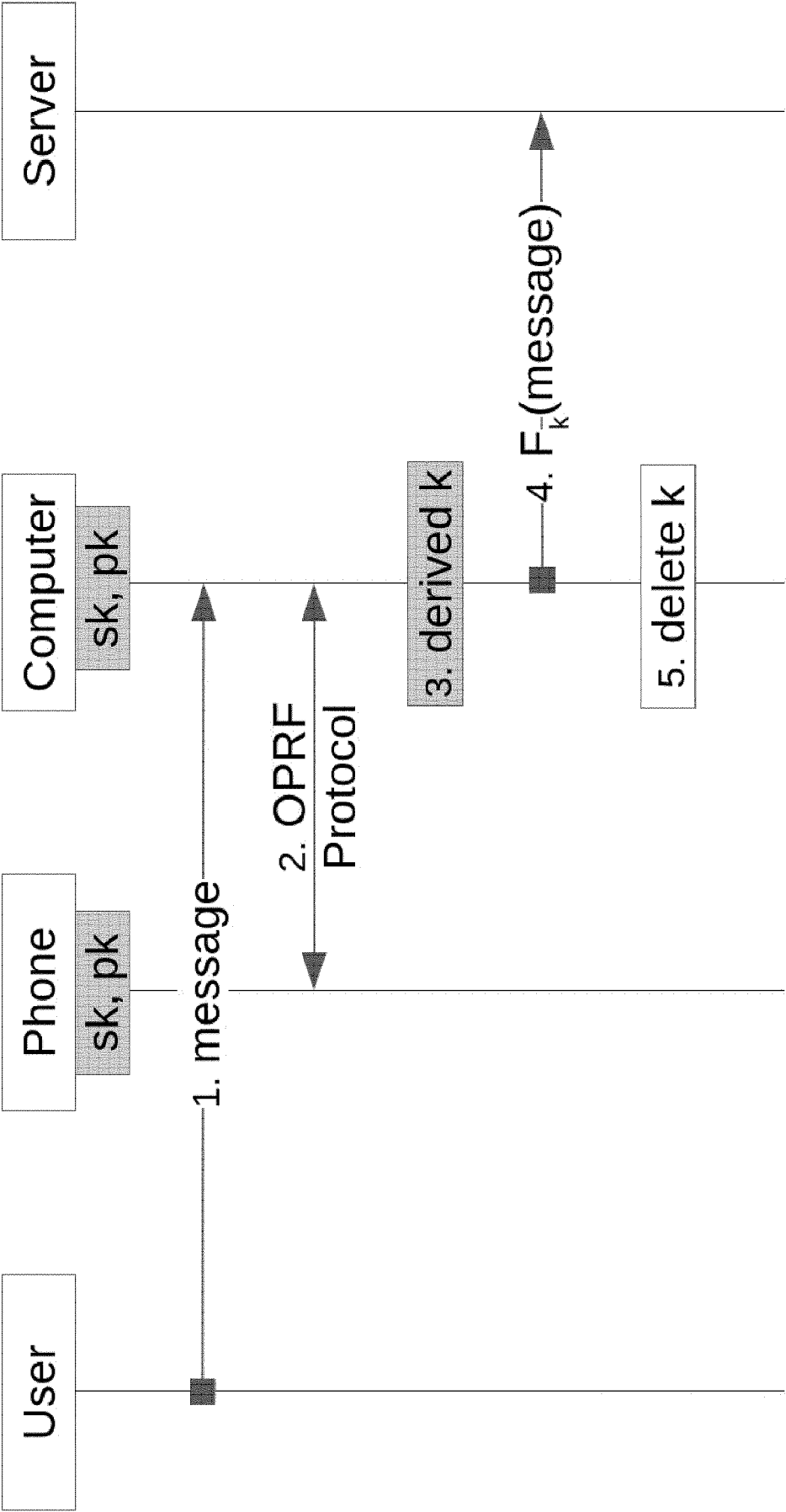


Fig. 2

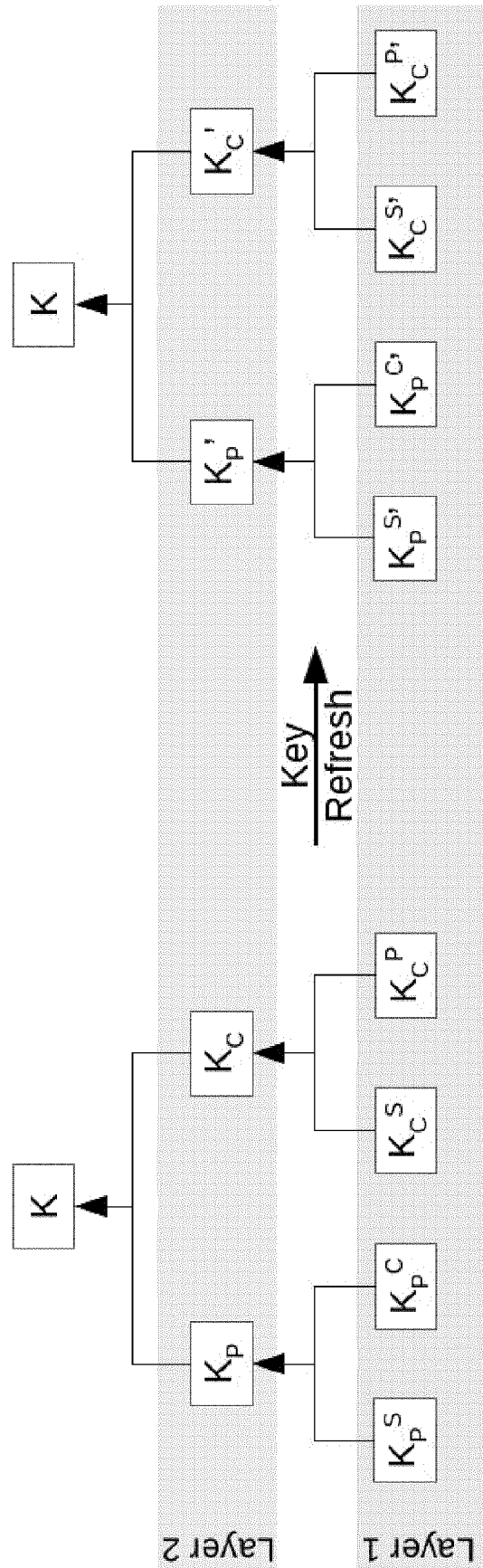


Fig. 3

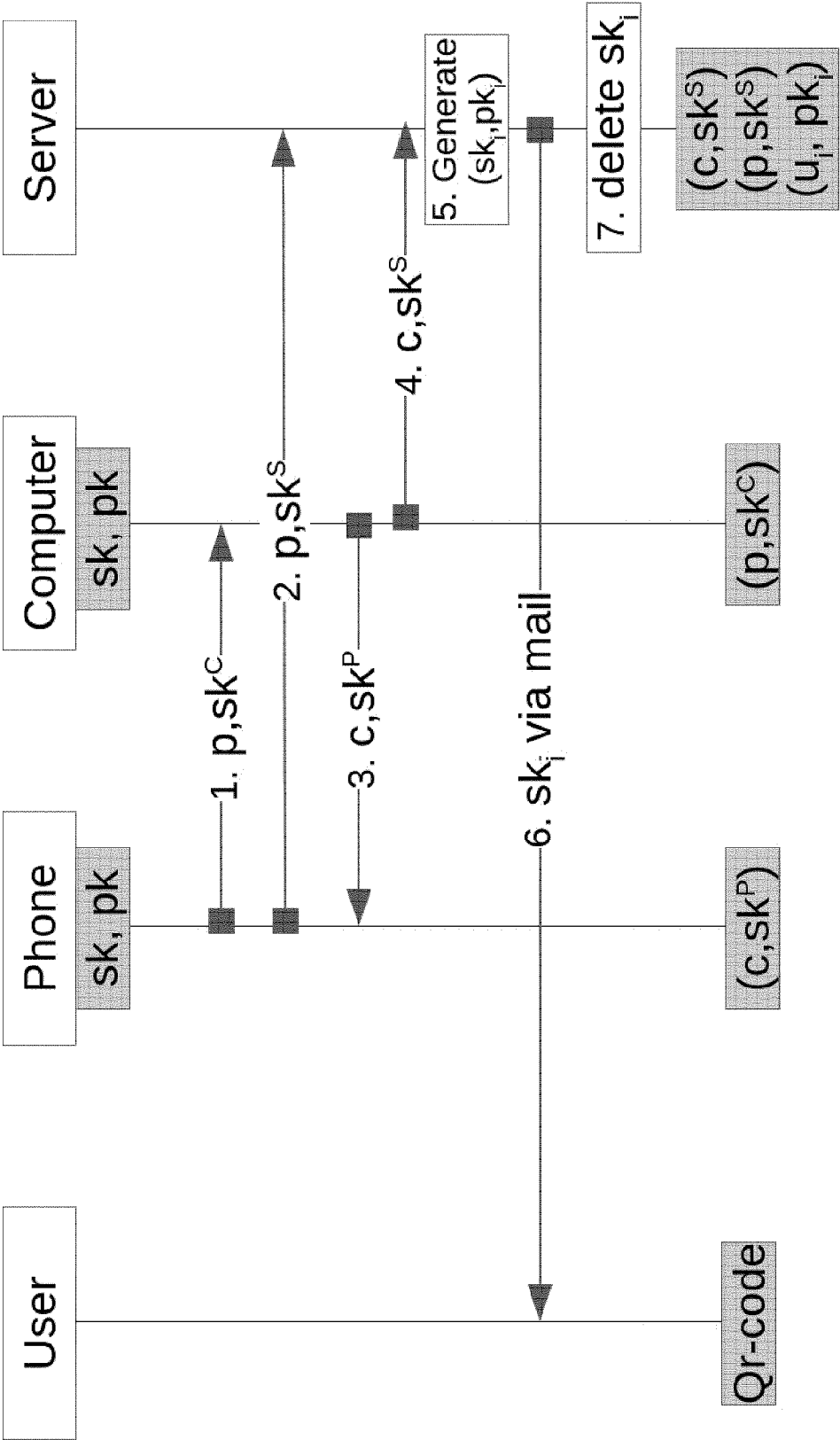


Fig. 4

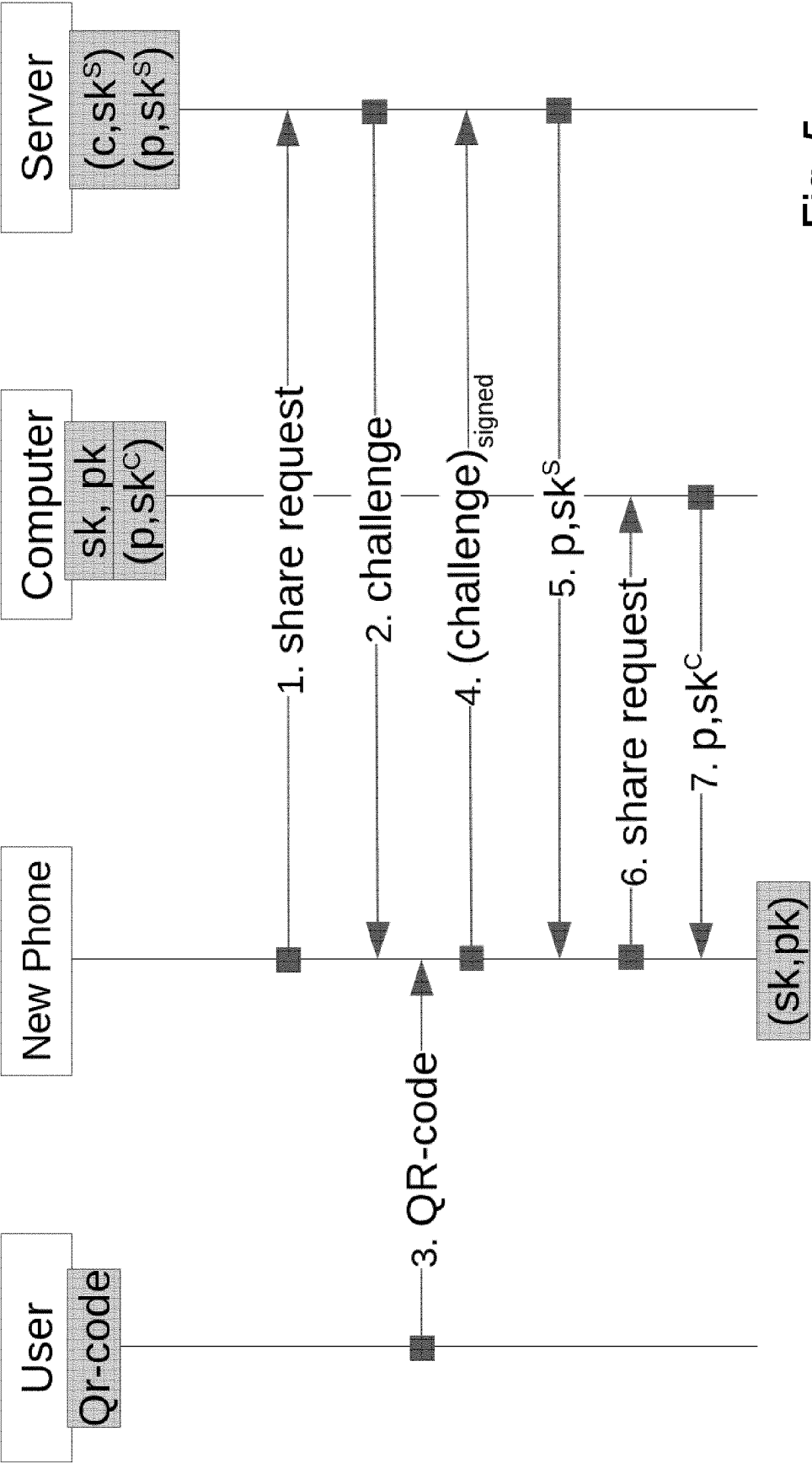


Fig. 5

Register

Username

admin

E-Mail Address

admin@domain.com

Password

••••••••••

Show

☒ I understand that I should keep the QR-code safe and private, as it holds the recovery credentials.

Perform the following steps:

1) Save the QR-code as image for future use.

2) submit this registration form.



Register

Fig. 6

Welcome admin

Visit the address below from you phone. Make sure you can connect to the server.
<http://192.168.1.17:8080>

Local Key Generated

Logs:

Key:

834fe9e205aae11cee6ace37dd15953c90c444af5329c9ae75df885afcf40ceb5fab017325e216e9daa2720b5534148921116b579add165f
e24b5feb7d97768c

Two shares computed.

Share Distribution completed.

Share Received from other device.

Generate New Keys

Fig. 7

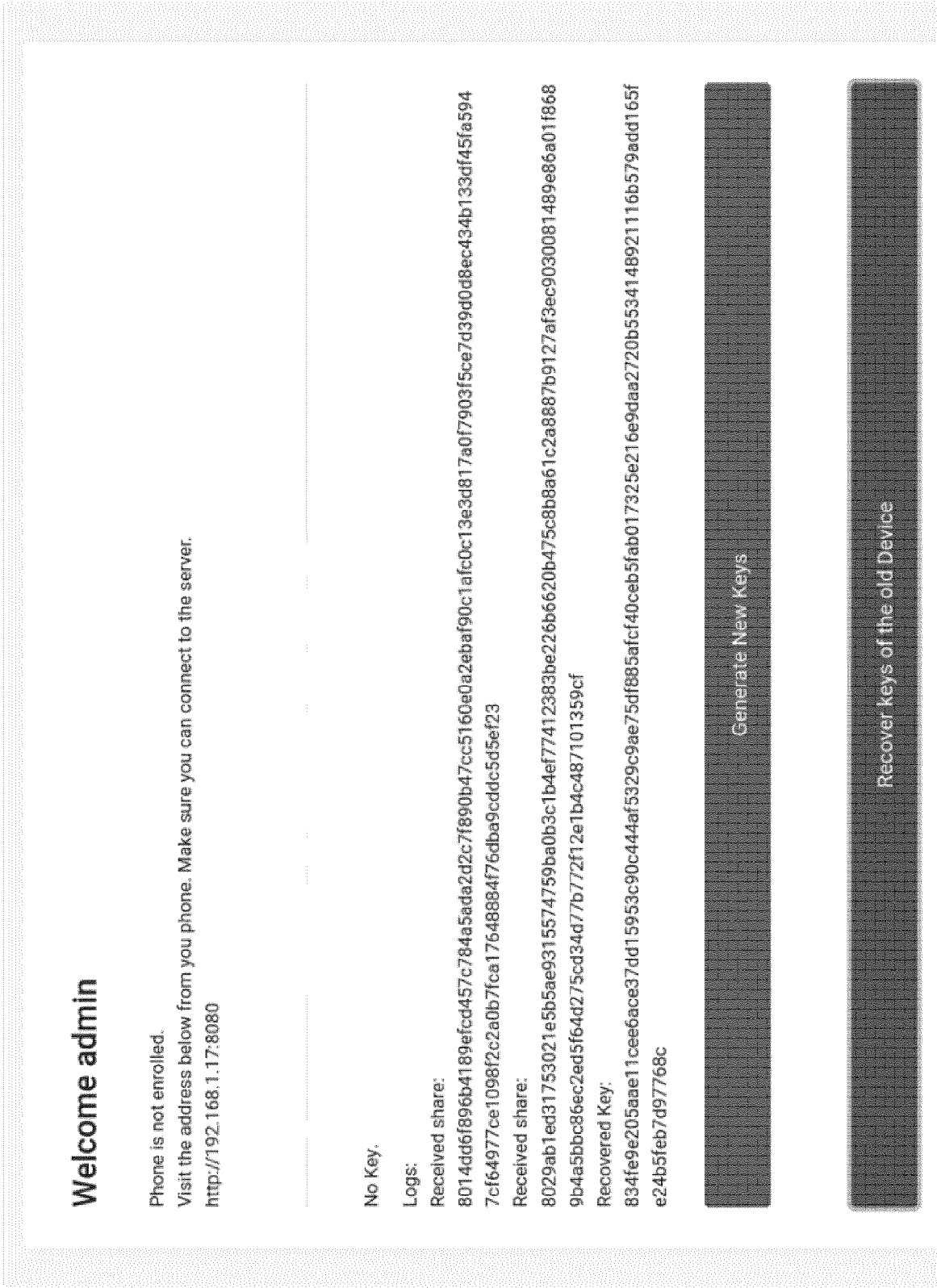


Fig. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2021/069669

A. CLASSIFICATION OF SUBJECT MATTER INV. H04L9/08 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04L		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ANDERS DALSKOV ET AL: "2FE: Two-Factor Encryption for Cloud Storage", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 27 October 2020 (2020-10-27), XP081800806, cited in the application	1, 9, 11-15
Y	pages 5-8; figures 1-6 ----- -/--	2-8, 10
☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents :		
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family	
Date of the actual completion of the international search 17 December 2021	Date of mailing of the international search report 04/01/2022	
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Horbach, Christian	

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2021/069669

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	DEVRIŠ İSLER ET AL: "Distributed Single Password Protocol Framework", IACR, INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH , vol. 20181015:121949 12 October 2018 (2018-10-12), pages 1-24, XP061026535, Retrieved from the Internet: URL:http://eprint.iacr.org/2018/976.pdf [retrieved on 2018-10-12] pages 10-12; figures 3(c), 3(d) -----	2, 3, 5-8, 10
Y	HORANDNER FELIX ET AL: "Horcruxes for Everyone - A Framework for Key-Loss Recovery by Splitting Trust", 2019 18TH IEEE INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS/13TH IEEE INTERNATIONAL CONFERENCE ON BIG DATA SCIENCE AND ENGINEERING (TRUSTCOM/BIGDATA), IEEE, 5 August 2019 (2019-08-05), pages 50-57, XP033653672, DOI: 10.1109/TRUSTCOM/BIGDATA.2019.00017 [retrieved on 2019-10-29] page 1 -----	4