

SECURE AUTHENTICATION IN A MULTI-PARTY SYSTEM

FIELD OF INVENTION

The present invention relates to authentication. More particularly it relates to securing and simplifying multi-level authentication in a multi-party system.

BACKGROUND

Internet authentication is currently based on passwords for authentication, even though passwords and password systems are insecure, easy to crack, guess, and subvert. Password security relies on 1) users remembering their password and 2) attackers not gaining access to the password. Since more secure passwords are hard to remember, common passwords like "12345" are used by huge percentages of the users on the Internet. Other solutions have been tried to increase the security of the system including One Time Passwords (OTP), which use an out of band method (typically the user typing in a code or a setup process taken before the token generator is shipped) to create one time tokens that are time based. Until recently OTPs have been generated by hardware tokens and were expensive to supply to users. Recent improvements have allows OTP to be distributed as software applications on mobile devices which are initialized by the user filling out some seed numbers. Although OTP provide protections from guessable passwords and brute force attacks they are still susceptible to a number of attacks like being re-used within the window and stolen (if the seed material is stolen the OTP can be generated by anyone). Another stronger security option is Public Key Infrastructures (PKI) which rely on public/private key pairs which is a huge benefit as the "secret" data (i.e. the private key) never leaves the user's control; but PKI systems are typically so complicated to implement that they are only used in huge organizations and cause numerous overhead problems like key revocation, provisioning each user, and managing keys when they expire etc.

In addition to increasing the security of authentication a better authentication system should make it easier for the user. Common security improvements like OTP, PKI, using additional security questions, making passwords longer, etc. make it more complicated on the user. Ideally, users

would both be able to use the system in a simplified manner but would also be given control over their authentication data to prevent fraud and to select different strengths of authentication based on their own preferences. With the expanded capabilities of smart phones and other user devices, it is now possible to transfer session/authentication data between channels, for example between the browser and the smart phone, this can enable stronger security authentication.

SUMMARY OF INVENTION

The invention is defined in the independent claims. Some optional features of the invention are defined in the dependent claims.

According to one exemplary technique, a method of authenticating a network user to another network entity, such as a network service provider, e.g. a bank or merchant, or another network user, includes executing, on a first user operated device, a first program to receive user inputted validation information, e.g. including a password, other user knowledge based data, a token and/or user biometric data. For example, the validation information may be inputted by the user using a touch pad or keyboard, and/or sensors, such as a camera or microphone, which are part of the first user operated device. The first program may store the input validation information together with a user credential, such as another password, token, biometric data or cryptographic key, e.g. a private key, of a private/public key pair of the user, on the first user operated device. The first program is further executed to direct transmission of the received validation data, and optionally the user credential, to an authentication server via the network.

A second program is executed on a second user operated device to receive information, most typically and for purposes of the description below a random number, which preferably serves as a session identifier, from another network entity via the network.

It should be recognized that the first user operated device could, for example, be a smart phone, tablet computer or other smart mobile communications device capable of communicating over the applicable network, such as the Internet, and the first program could be an app executable on such smart mobile device, or a library embedded within another program. Alternatively, the first user operated device could be a personal computer, laptop or other less mobile processing device capable of

communicating over the applicable network, and the first program could be an application, e.g. a browser pop-up or separate authentication application, executable on such processing device. Likewise, the second user operated device could be a smart mobile device or less mobile processing device and the second program could be an app, e.g. a browser app, such as Safari™, or a browser application, such as Firefox™. Thus, the first user operated device and the second user operated device may be different devices, e.g. a smart phone and personal computer, or the same device, e.g. a laptop computer.

The first program is further executed to receive an input transferring the random number received by the second program, to the first program, and to direct transmission of the transferred random number to the authentication server via the network. In alternate embodiments the random number and login initiation request may come directly from the authentication server, bypassing the need for the random number to be transferred between the second program and the first program. In response, an identifier of the other network entity, other information, and authentication policy requirements of the other network entity are received from the authentication server via the network.

With regard to the transfer of the received random number to the first program, in some implementations it may be preferable for the second program to be further executed to direct a presentation of the received random number on a display screen communicatively connected to the second user operated device. In such a case, the received random number could be in the form of an optical code, e.g. a QR code, and the input transferring the received first random number to the first program could be received from a camera included in the first user operated device. For example, the input transferring the received random number to the first program might be a digital photograph of the presented optical code. In other implementations it may be preferable for the second program to be further executed to direct printing, on a printer communicatively connected to the second user operated device, of the received random number. In such a case, the input transferring the received first random number to the first program could, for example, be a digital photograph of the printed optical

code. In still other implementations, the random number could be transferred using other techniques, for example by an audio code or near field radio communication.

After receiving the authentication policy requirement of the other
5 network entity, the first program directs transmission of the input validation information corresponding to the received policy requirements to the authentication server via the network. That is, the first program determines the validation information necessary under the other network entity authentication policy requirements, retrieves this validation information from
10 storage and/or receives the information via inputs from the user, and directs transmission of this information to the authentication server.

If the authentication server is able to validate the user based on the transmitted validation information, a request for a user credential is received from the authentication server via the network. The first program signs a
15 message, which includes the transferred random number and the received other information, with the stored user credential, and directs transmission of the signed message to the authentication server via the network to authenticate the user.

It should be understood that the authentication server validates, based
20 on the validation information received from the user, that the user is who he/she/it says he/she/it is. Thus, this validation is a form of authentication and could, for example, be characterized as an initial authentication, since authentication may not be fully completed until the user is further authenticated based on the user credential.

25 It will also be recognized that if the credential is a key of a private/public key pair of the user, then the first program is preferably further executed to generate a private/public key pair for the user, typically during initialization. In such a case, the stored user credential is the private key of the generated user private/public key pair and the credential optionally
30 directed to be transmitted to the authentication server is the public key of that key pair. Thus, the authentication server can complete the authentication by applying the transmitted user public key to the transmitted signed message to recover the random number and the other information.

Whether or not the credential is a user key or some other credential, after transmission of the signed message has been directed, the second program is further executed to receive from the authentication server, or the other network entity, or both, via the network, an indication that the user has
5 been successfully authenticated to the other network entity.

It should also be understood that the user may have a single credential which is used for authentication to multiple other network entities or a different credential for authenticating to each of multiple other network entities. If different credentials are used, the first program is further executed to store
10 multiple credentials, potentially one for each account at each provider, and to use the correct credential when accessing the specific account or provider services. That is, if different credentials are used, the first program is further executed to store another user credential on the first user operated device, and to optionally direct transmission of the other credential to the
15 authentication server via the network.

The second program is further executed to receive another random number, which will be referred to below in this section as a second random number, from a different other network entity, which will be referred to below in this section as a second other network entity, via the network, and the first
20 program is further executed to receive an input transferring this second random number to the first program.

The first program also directs transmission of the transferred second random number to the authentication server via the network. In response, it receives an identifier of the second other network entity, further other
25 information, which will be referred to below in this section as second other information, and authentication policy requirements of the second other network entity, from the authentication server via the network.

The first program directs transmission of the stored and/or newly input validation information corresponding to the received second other network
30 entity authentication policy requirements to the authentication server via the network. If the authentication server is able to validate the user based on the transmitted validation information, the first program receives a request for a user credential from the authentication server via the network.

The first program signs another message, which includes the second random number received from the second other network entity and the second other information received from the authentication server, with the stored other credential, and directs transmission of the signed other message to the authentication server via the network. The second program is further
5 executed to receive an indication that the user has been successfully authenticated to the second other network entity, from either the authentication server or the second other network entity, or both, via the network.

According to other preferred examples, the first program can be further executed to generate user secret data, such as an extended password or hash of a password, and divide the
10 generated secret data into multiple portions, including at least a first portion and a second portion. In such a case, the first program also encrypts the user credential with the secret data, i.e. the full secret data including both the first and second portions, and the stored user credential described above will be the encrypted credential. The first program additionally directs transmission of the second portion of secret data to the authentication server via the
15 network. During later access, if the authentication server is able to validate the user based on the transmitted validation information, it transmits the second portion of secret data, which was previously transmitted to it, back to the first program. Accordingly, the first program also receives, typically with the request for the credential, the second portion of secret data from the authentication server via the network. It then combines the first portion of secret data with the
20 received second portion of secret data, and decrypts the stored encrypted credential with the combined portions of secret data, i.e. with the full secret data. Thus, if such a split secret is utilized, the above described signed message would be signed with the decrypted user credential.

Advantageously, the first program is further executed to receive user inputted user
25 authentication policy requirements and to optionally store such requirements on the first user operated device and transmit those requirements to authentication server. In such a case, after receiving the authentication policy requirements of the other network entity, the

authentication server determines any additional authentication policy requirements based on any differences between the user authentication policy requirements and the other network entity authentication policy requirements. The authentication server transmits the determined additional authentication policy requirements to the first program together with the other entity authentication policy requirements. Thus, in such a case, the first program receives the authentication policy requirements of the other network entity and any determined additional authentication policy requirements. It also directs transmission of the user input and/or stored validation information corresponding to the received additional authentication policy requirements, together with that corresponding to the other network entity authentication policy requirements, to the authentication server via the network. That is, the first program directs transmission, to the authentication server, of the user validation information necessary to meet both the other network entity and the user authentication policy requirements.

In accordance with other preferred examples, the authentication server may receive notice that the first user operated device is no longer in use, or has been lost or stolen, or that the users credential(s) has/have otherwise been compromised. If so, the second program can be further executed to receive another random number, which will be referred to below in this section as a third random number, from the other network entity via the network. That is, this third random number is received from the same network entity as the received random number first mentioned above.

The first program, which may now be executing on a third user operated device, such as a new smart phone, receives an input transferring this third random number received by the second program, to the first program, and directs transmission of the transferred third random number to the authentication server via the network.

In response, the first program receives an identifier of the other network entity, still further other information, which will be referred to below in this section as third other information, and authentication policy requirements of the same other network entity from the authentication server via the network. The first program directs transmission of validation information corresponding

to the received other network entity authentication policy requirements to the authentication server via the network. If the authentication server is able to validate the user based on the transmitted validation information, the first program receives, from the authentication server via the network, a
5 notification that the user has been validated but cannot be authenticated because the user credential has been invalidated.

Advantageously, the second program can be further executed to receive a redirection instruction from the authentication server via the network, which redirects the user to the other network entity's reenrollment
10 website on the network.

Additionally or alternatively, the first program can be further executed to receive a request for a replacement credential from the authentication server via the network. In response, the first program generates a replacement credential, stores the generated replacement credential on the device
15 currently being operated by the user, and optionally directs transmission of the generated replacement credential to the authentication server via the network. If the replacement credential is transmitted to the authentication server, the authentication server can also then send, via the network, the replacement credential and a certificate of validity of the replacement credential to the
20 other network entity for use in re-enrolling the user with the other network entity.

Beneficially, after directing transmission of the generated replacement credential to the authentication server, the first program can be further executed on the currently operated device to direct transmission, to the
25 authentication server via the network, of another message, which includes the third random number and the third other information, signed with the replacement credential to authenticate the user.

To enroll the user with another network other entity, the first program can be executed on the first user operated device, such as a smart phone,
30 tablet computer or other smart mobile communications device, to cause the user device to receive a set of user identities and enrollment data (e.g. name, date of birth, etc.), and to store the received set of user identities such that each identity is associated with a respective set of enrollment data. The

second program directs transmission of an enrollment request to the other network entity. In response, the second program receives, from the other entity via the network, a random number. The first program is executed to cause the first user device to receive an input transferring the random number received by the second program to the first program, and to transmit the transferred random number to the authentication server via the network. In response, the first user device receives an identifier of the other network entity and enrollment data requirements of the other network entity from the authentication server via the network.

Optionally, during enrollment, the first user device receives a user input selecting one of the stored set of user identities. In response to receipt of the selection, the first program causes the first user device to automatically retrieve the stored enrollment data associated with the selected user identity, and select data from the retrieved enrollment data that corresponds to the received enrollment data requirements. The user may be given the choice to edit the selected data. The first user device then transmits the selected, optionally edited, data to the authentication server.

The first program also causes the first user device to transmit the required validation information to the authentication server along with a new credential for use with the other network entity. The credential may be established by the first user device in various ways, as is well understood in the art. For example, the first program may generate the credential. In those cases in which the credential is a public/private key pair, the first user device generates or receives the public/private keys, and transmits the public key to the authentication server together with a request that the authentication server sign the public key to create a certificate. The authentication server then transmits the certificate back to the first user device. Accordingly, it should be recognized that the authentication server can serve as both an authentication server and a certificate authority, and the first user device obtains both user certificates and user authentications from the same entity, i.e. the authentication server.

In accordance with still other examples, the first program is executed to cause the first user device to notify the user of a transaction

between the user and another network entity, e.g. a service provider such as a merchant or bank, with which the user has already enrolled or otherwise has an existing relationship. To do so, the first user device receives, from the authentication server via the network, a transaction identifier, transaction approval and authentication requirements, and a message regarding the transaction, which could for example include the transaction details. The message is encrypted with a credential of the user. If the credential is a public key of a private/public key pair of the user, the private key of the user private/public key pair is known only to the applicable user. The message is also signed with a private key of a private/public key pair of the other network entity. The public key of the other network entity private/public key pair is known to the first program and hence to the user.

In response, the first program is further executed to cause the first user device to validate and decrypt the message by applying the public key of the other network entity and the private key of the user to the received signed encrypted message. The first program then directs the first user device to present, e.g. display, the validated decrypted message to the user, obtaining authentication information, such as validation information, as needed. That is, the first user device receives a user input of any authentication information that is required and otherwise not available, e.g. previously stored, on the first user device. The first user device next receives user input representing a transaction approval. The first user device then transmits the input transaction approval, or the input validation information and/or other authentication information such as a credential, or both, depending is what is required, to the authentication server via the network, in order to authenticate the user and/or approve the transaction.

Additional objects, advantages, and novel features of the present invention will become apparent to those skilled in the art from this disclosure, including the following detailed description, as well as by practice of the invention. While the invention is described below with reference to particular embodiment(s), it should be understood that the invention is not limited thereto. Those of ordinary skill in the art having access to the teachings herein will recognize additional implementations, modifications, and embodiments, as

well as other fields of use, which are within the scope of the invention as disclosed and claimed herein and with respect to which the invention could be of significant utility.

BRIEF DESCRIPTION OF DRAWINGS

Each of these drawings represents an example embodiment of the systems they are depicting.

FIG 1. Illustrates the main architectural components.

FIG 2. Illustrates an example login screen.

FIG 3. Illustrates the communication between the user (both browser and mobile phone), the authentication server, and the network provider.

FIG 4. Illustrates the primary subsystems on the mobile user device.

FIG 5. Illustrates the primary subsystems on the authentication server.

FIG 6. Illustrates the split key system and the high-level data flow.

FIG 7. Illustrates the sample interactions between the use, the authentication server, and the provider.

FIG 8. Illustrates the data control areas for various pieces of data within the system.

FIG 9. Illustrates the communication between the user (both browser and mobile phone), the authentication server, and the network provider.

FIG 10. Illustrates a sample paper based enrollment process.

FIG 11. Illustrates a sample communication flow for the notification system.

DETAILED DESCRIPTION

OVERVIEW

First Authentication Framework

There may be provided an authentication framework, see FIG 1, that includes a mobile device of the user which has a network connection, e.g. a smart phone 400 or tablet, and a software application capable of executing various cryptographic and authentication operations¹⁰¹. A network service 103, like a web site or terminal server, is also included. Access software is used to reach the network service, like a web browser 102, that may reside on the mobile device or on a secondary device like a desktop or laptop. The access software may have a trusted component which can reliably send security data to the user's mobile device

including information to validate that credentials being provided by the user are for the current network service. The trusted component of the access software may be a browser plug-in. Additional security information, provided by the trusted component, may include the network service's address or URL
5 (in the case of a web service). An authentication server 100 is attached to a network and reachable by the network service, the access software and the mobile device.

A first transfer connection 106 between the access software and the mobile device software, like a scanned QR code or near field communication
10 system, is used to transfer an authentication session identifier (ID), and may include additional information for session startup. The first transfer connection may be augmented by a more complete communication channel (for example Bluetooth, near field radio, USB connection, etc.) allowing validation between the access software and the mobile software application that they are
15 connecting to the same network service. The session ID received on the first transfer connection may be transferred via optical display of a QR code. If so, evaluating the session ID may include reading the QR code with a camera and decoding the QR code.

A second transfer connection 107 between the mobile device software
20 and the authentication server that performs the transfer of security checks for liveness of the session, and checks credentials of the user and the network service. The authentication server may include a fraud and anomaly detection system 504. The credentials of the user may be based on an asymmetric public key cryptography using a public/private key system and the credentials of the
25 user may be specific to the network service being accessed.

If a public/private key system is used during authentication, it may include a set of communication keys/certificates which allows authentication between the authentication server and the network service and between the authentication server and user mobile software. It may also include a set of
30 user validation keys/certificates which allows authentication between the user and the network service. The set of user validation keys may also include user-to-user validation keys, and a single key/certificate for use across multiple services, or a unique key for every user-service pairing.

Additionally, the authentication server could be used to validate authentication data, and determine the process for dealing with expired, lost or otherwise invalid certificates. The authentication server may also perform fraud detection and auditing. Finally, one or more certificate signing
5 authorities could be included and used by the authentication server to approve certificates. Preferably, three signing certificate authorities would be used: one for network service credentials, one for user to network service credentials, and one for user to authentication server credentials. The process for handling lost credentials may include invalidating the lost
10 credentials.

The security checks may further include additional authentication factors such as biometric data, token based authentication data, or additional knowledge based factors. If biometric data is one of the factors, it could include hand recognition using a camera on the device, biometric data
15 collected from devices attached to the user mobile device, fingerprint recognition using a USB fingerprint scanner. The additional authentication factor(s) used could include knowledge based question(s), such as a password and/or personal identification number (PIN), or a verifiable property like a pre-registered phone number. The additional authentication factor(s)
20 used could include tokens either on or attached to the mobile device, such as a smart card read through an attached card reader, a secure data token accessed through near-field radio, and/or data read from the SIM card of the mobile device.

A third transfer connection to the access software allows the access
25 software to know when the user authentication has been completed, and may be between the access software and either the authentication service 105 or the network service 108. The access software may automatically transfer to a logged in state when the authentication is complete.

Second Authentication Framework

30 This authentication framework includes a variety of services and users of those services across a network, which have a variety of credentials allowing users to communicate. The users could be automated processes instead of people. Also included are authentication data indicating which user-

service credentials are valid, lost, or expired and usable to determine if logins are allowed. A system is provided for communicating between parties, and may include traditional network protocols, proxies, or reverse proxies.

One or more hierarchical authentication server(s) are also provided,
5 with the top tiers of the hierarchical authentication server(s) capable of redirecting individual requests for authentication to lower levels or tiers based on the services being authenticated. Each authentication server, using the authentication data, can make determinations as to if user credentials are still valid. The hierarchical authentication server(s) may be a single level of server
10 where one server or cluster of servers handles authentication requests. A lower tier server could, if desired, reside behind a firewall or perimeter network gateway and handle services on the local or organizational network. Logins may, for example, be denied because credentials have been marked as lost and so are no longer valid.

15 Third Authentication Framework

This authentication framework includes an authentication server attached to a network and reachable network service, like a website or terminal server. Also included is a set of user credentials on the authentication server, some of which may be marked as invalid because, for
20 example, the device they resided in was lost.

Access software is used to reach the network service. A communication system between the network server and the authentication server tells the network service that the user is valid but the individual credentials are invalid or do not exist. The communication system may be a
25 method query within an API object returned from the authentication server or a custom protocol that returns the validity of the credentials.

A simplified method is implemented to redirect the access software to an enrollment or alert page for additional validation. The simplified redirection method may be a web redirect message to an enrollment page, or an inter-
30 application communication, like an Android intent to start a custom application for enrollment. The user may be redirected because they need to enroll on the site or to re-enroll on the site, in which case the user enrollment information could be filled in with pre-configured identity information as

approved by the user. See FIG 8. User identity information may be stored on a user mobile device 806, or on the authentication server. The identity information may be pre-grouped into sets of identity allowing the user to have multiple-identities (business, personal, etc.) and select between them.

5 Data Securing System

Also provided is a system for securing data on a mobile device that includes an encrypted data storage location (keystore or datastore) which uses a symmetric key for encryption and decryption, see FIG 6. User entered secret data is divided into multiple parts 601, where one portion of the user entered secret data is used locally to obtain the first portion of the symmetric decryption key 605. The user entered secret data may be a password or passphrase, a biometric template or match, or an image drawn by the user. A network reachable server uses a second portion of the user entered secret data to retrieve the second portion of the symmetric key for decryption 604. For example, half of the password may be combined with system level data to obtain a half of the decryption key, such as by using that half of the password to decrypt a local block of data containing the half of the decryption key. A fraud detection policy on the network server 603, which may for example limit the number of failed login attempts, alarms or responds to suspicious requests for the second portion of the decryption key.

20 Authentication Server

See FIG 7, also provided is an authentication server 713 that includes a set of data relating to each user specifying which services they have credentials with, what type of authentication is required for access, and if the credentials are valid. A minimum security policy is specified by each of the services using the authentication server 711. Policy management software 700 allows users to update their own records to (i) modify the types of authentication required to access a particular service or set of services as long as it meets the minimum specifications of the service, (ii) delete their credentials, or (iv) otherwise interact with their user data. The policy management software may reside on the users mobile device or on a web site accessed by the user. The modification could include adding a new factor of authentication (for example specifying that login requires a face as well as a

PIN), or changing the factor to be more accurate or restrictive (for example use face recognition instead of a hand recognition).

Fourth Authentication Framework

5 An authentication framework may include a mobile device of the user which includes a network connection and a software application capable of executing various cryptographic and authentication operations. A network service, like a web site or terminal server, is also included. Access software is used to reach the network service, like a web browser, that may reside on the mobile device or on a secondary device like a desktop or laptop. An authentication server
10 attached to a network and reachable by the network service, the access software and the mobile device. If desired, the authentication server can include a fraud and anomaly detection system. See FIG. 9. An identifier 902 is supplied to the browser and uniquely identifies the mobile device's software application. The identifier could be a unique value per user like a phone number or user name.

15 A first connection between the mobile device software application and the authentication server performs the transfer of security checks for liveness of the session, and credentials of the user and the network service. The credentials of the user may be based on asymmetric public key cryptography using a public/private key system, and the credentials could be specific to the network service being accessed. The security checks may further include additional authentication factors including: biometric data, token based authentication data, or additional
20 knowledge based factors. The biometric data could be collected from devices attached to the user mobile device

25 A second connection to the access software allows the access software to know when the authentication has been completed. The second communication channel may be between the access software and the authentication service, or between the access software and the network service.

Fifth Authentication Framework

This authentication framework includes a mobile device of the user which has a network connection and a software application capable of

executing various cryptographic and authentication operations. A network service, like a web site or terminal server, is also included. Access software is used to reach the network service, like a web browser, and may reside on the mobile device or on a secondary device like a desktop or laptop. An authentication server is attached to a network and reachable by the network service, the access software and the mobile device. Means are provided to uniquely tie the browser session to the mobile device's software application. These means may be communicated from the browser to the software application via an optical code, such as via optical display of a QR code. Processing this communication could include reading the QR code with a camera and decoding the QR code.

A first connection between the mobile device software application and the authentication server performs the transfer of security checks, credentials of the user and the network service, and provides a secure messaging channel. The secure messaging channel may be (i) a unidirectional communication means from the network service to the mobile device software application, or (ii) a bidirectional communication means for the network service to query additional information from the mobile device software application. The secure messaging channel may have a defined messaging protocol which includes the ability to transmit one or more of encrypted message data, approval requests, or policy information.

A second connection to the access software allows the access software to know when the authentication has been completed.

Authentication Enrollment System

An authentication enrollment system includes a mobile device of the user which has a network connection and a software application capable of executing various cryptographic and authentication operations. See FIG 10. A printed document 1000 related to an individual account related to a network service, which includes the code 1008 to uniquely identify the enrollment session, is also included. An authentication server is attached to a network and reachable by the mobile device software application and the network service. Means are included to uniquely identify the enrollment session to the mobile device's software application. The means may be communicated from

the printed document to the software application via an optical code including a QR code, processing the communication may include reading the QR code with a camera and decoding the QR code.

A first connection between the mobile device software application and the authentication server that performs the transfer of security checks and authentication of the user. A second connection to the network service allows the network service to know when the enrollment has been completed.

DESCRIPTION OF EMBODIMENTS

Multiparty Authentication System

A multi-party authentication system which uses an essentially un-trusted authentication provider 100 to validate users to network service providers 103. The system involves a trusted user device, like a mobile tablet or smart phone 400, that secures and holds the various user credentials (for example public/private keys and certificates); and can collect a variable number and type of authentication credentials including biometric, knowledge based (i.e. passwords), and token based. The authentication service 100 provides user driven provisioning and controls allowing real-time scalable authentication across a WAN and allows lost key recovery and simplified enrollment or re-enrollment.

Reference is now made in detail to the description of the embodiments as illustrated in the drawings. While embodiments are described in connection with the drawings and related descriptions, there is no intent to limit the scope to the embodiments disclosed herein. On the contrary, the intent is to cover all alternatives, modifications and equivalents. Those of ordinary skill in the art will appreciate that other embodiments, including additional devices, or combinations of illustrated devices, may be added to, or combined, without limiting the scope to the embodiments disclosed herein.

The techniques disclosed herein may be based on a multi-party system (the user, the authentication service provider 100, and the network service provider 103) and includes the following main architectural components (see FIGS 1 through 5 for views of the architectural components):

- A user mobile device 400 which includes a network connection 402, data input system 401, and a software authentication application or app

(Qapp) 101 which allows various authentication functions to be done. The Qapp 101 manages the authentication communication 409, collects extra authentication data 403 and 404, and stores user credentials and data 405 through 408. User mobile devices can be any personal device including smart-phones, tablet computers, laptop, etc. One embodiment is to use a Smart phone which includes both cellular based networking and WiFi networking. A second embodiment of a mobile user device might be a specialized authentication device. Ideally the user mobile device also includes additional sensors 404 (like a camera, microphone, etc.) which can be used to collect biometric 404 or additional token data 403.

- A network service provider 103 (Provider), which has information that the user is required to login to access. The Provider's main role during authentication is to map users to the correct Provider account. Optionally the Provider can re-verify authentic credentials and perform other security checks. One embodiment is a web site 103 like an eBanking web site which allows the user to login to gain information about their account. A second embodiment might be a network enabled desktop login (like a Windows login screen).
- A user's network service access software. This is the software the user interacts with to login via communications channel 108. For web sites the access software is the web-browser 102, other types of network services might have specialized access portals. Ideally, it would have the ability to transfer a set of data between the access software and the authentication application on the mobile device. See FIG 2. One embodiment is to show a visual QR code 200, a 2 dimension bar code system, other options include audio codes, near field radio communications, etc. For the remaining description we have assumed a QR code is used for transferring information in the first channel 106 (i.e. between the access software (i.e. browser) and the authentication application). In at least some implementations, the network service should be capable of collecting a unique user ID. See FIG 9. Sample embodiments might include i) the user directly entering the data into a

field on the web site 902, ii) the ID being stored in a cookie or other browser storage location, or iii) from a communication means with the user's mobile device (attached USB, nearfield, etc.) where the user's mobile device then provides the unique user ID.

- An authentication server 100 providing a variety of services (Qserver). The Qserver acts as an intermediary PKI management portal ensuring that various user and Provider credentials are correctly setup and used including 501 and 505; it handles additional forms of authentication 502 including biometric and token based options; it allows users to manage their own authentication including reporting lost phones or upgrading authentication through a policy mediation system 503; optionally, it manages user identities; and it does fraud detection 504 on authentication requests and authentication data used. In one embodiment the authentication server is a single Internet reachable host 500 or cluster of servers allowing network connected devices to connect, via communications channels 104, 105, and 107 and managed by 506, that requests or sends a variety of information. In a second embodiment the authentication server may consist of a hierarchy of servers allowing decision on authentication to flow between servers. In this embodiment the lower level server may provide authentication services to an organizational or corporate network behind firewalls or other network and security boundaries. In one embodiment the Qserver is housed on a server separate from the Provider to limit the security risks of losing authentication data.

Because the roles of each party spread the trust throughout the system, if one party is compromised risks to the entire system can be limited. See FIG 7. For example one strength of the techniques disclosed herein is that if the user losses their mobile device the credentials can easily be revoked 705, credentials may also require the user to enter secret data (like a password) to access – which can be audited, and secure sites will require biometric data to access. In addition, none of these controls requires Provider intervention or modifications. Another security benefit is that during phishing attacks the user authentication data is never transferred to the Provider (or pretend Provider)

so no credentials can be stolen. Finally, if the Qserver is compromised; because the Qserver does not have access to user private credentials, an attacker even with all of the data from the Qserver, can not become a user or gain any additional access to a Provider. Further, because the authentication
5 server acts as an authentication gateway, many events within the system can happen completely transparently or behind the scenes for the user. For example re-enrollment to a provider after a mobile-device is lost can happen without any addition interaction on the part of the user. It is also possible for one embodiment of the authentication server system to actually consist of a
10 set of authentication servers, some of which reside on organizational networks behind firewalls or other perimeter devices, and authentication requests are coordinated between the servers to decide based on the network service provider being accessed which specific authentication server should be responsible.

15 Figure 7 depicts the some of the user controls and management areas 714 and 715 possible. The first section 716 illustrates the policy mediation done at the authentication server 713 where user policies 700 for how much authentication is desired to access a particular provider communicated to the server over 710 are analyzed with the provider's own policies 701
20 communicated over 711 to create the ultimate requirements 702 for the user to access the provider's site. The next three sections 717, 718, and 719 respectively show that if the user wants to change their password 703, losses their credentials 705, or the credentials expire 707 it is not an activity that requires the provider be involved 709. The communication happens between
25 the authentication server 713 and the user control area 714 through 703 through 708.

Figure 8 illustrates the ability of the user app 800 to maintain multiple accounts across one or more providers 801 and multiple sets of identities 806. Account information also includes the users authentication requirements. The
30 user identities 806 may include multiple data records. The authentication server 713, as discussed in the policy management section, then mediates authentication requirements between the user and the provider for each account that the user has configured 813. The authentication sever 713 also

facilitates communication between multiple user areas and multiple provider areas 811, and 812. For example for the first user/provider pair information flows from the user area 714 through channel 807 to the first provider area 811 over channel 809. The second user/provider pair sends different data over a similar set of channels 808 and 810. The providers each maintain their own set of account data 802, and 804 including the authentication configuration options 803 and 805.

In addition to the architectural pieces there are a set of communications that happen between the different parties. One common interaction with the system is when a user logs into a web site. The following is a high-level view of one embodiment for login.

1. The user goes to a web site that implements the login system. FIG 2. shows a screen shot of what the login might look like. In addition to the normal account login selection a QR code 200 with the Qcode is displayed. The Qcode includes a header block (described below) and a Session Id (Qsid) which is guaranteed to be unique across all users for as long as the code is valid and acts as a simple identifier for the authentication session.

2. Referring now to Figure 3, the user, on their smart phone, starts up the Qapp and scans the Qcode 302. This starts a communication 303 with the Authentication Server (Qserver) 100 and, depending on the policies of the user and the Provider which are provided in 304 the user will enter additional authentication information 305 that may include a pin, token data, and/or biometric data. For the purposes of a higher security site, for example PC Banking, the user might go through the following. FIG 3 shows an example login.

i. They will see a message saying "Would you like to login to Provider X". This will allow the user to validate that who they are trying to login to is the same location that they are connecting to in 102.

ii. The user will approve the login and then enter their secret data and if the site 103 requests three-factor authentication they may also need to submit some form of biometric data (such as a picture of their face, hand image, or voice sample).

iii. The session information, session validation, approval, and the biometric data is then submitted in 306 to the Qserver server 100 for validation. Behind the scenes if the Qserver approves the authentication in 307 and then it will notify the Provider (the site can then do an independent validation of the user's challenge/response) and the users browser gets automatically refreshed and logged in.

3. The user is now logged in – besides scanning the Qcode in 302 they do not need to enter any information or even click on any links on the web site. This makes a virtually transparent, completely automated, and yet secure authentication system.

The following is a high-level view of another embodiment for login. See FIG 9.

1. The user goes 901 to a web site that implements the login system. Instead of the traditional username and password fields the web site has a single field for the user to enter their unique user ID 902. The unique ID could be assigned like a username or be something like a phone number. The user then clicks submit and the web site with collaboration from the Qserver assigns a unique session id (Qsid) 900 and starts authentication. In a second embodiment the unique ID might be saved in a cookie or retrieved from the mobile device during access. It is also possible to tie the browser and mobile device together by transmitting from the browser the Qsid or other identifier specific to the session. The transmission could be in the form of a QR code displayed to the user and scanned via the mobile device.
2. The provider communicates the unique user ID and the Qsid to the Authentication Server 903.
3. Based on the unique user ID the authentication server contacts the user's mobile device 904. This communication with the Authentication Server (Qserver) which will include the requirements for authentication 905. Depending on the policies of the user and the Provider the user will enter additional authentication information that may include a pin, token data, and/or biometric data 906. FIG 9., which shows an example

login as in FIG. 3 described above, also shows the completion steps 907 and 908 to finish the login.

4. The user is now logged in – besides providing their unique user ID, they do not need to enter any information or even click on any links on the web site. This makes a virtually transparent, completely automated, and yet secure authentication system.

Specific Security and Architectural Issues

The Qcode includes the following information: a header tag which specifies that this is an authentication token usable with the techniques disclosed herein, and a Qsid which is long enough to make guessing virtually impossible, guarantee uniqueness within the environment, and long enough to make reuse of the Qsid happen infrequently. In one embodiment, the Qsid is a random number at least 128- bits long. Optionally, the Qcode may include session specific information like the type of login/enrollment requested, or alternate authentication server.

In addition to the Qsid, there is a liveness ID (Qliveness) which is another piece of information passed from the Provider, through the Qserver to the Qapp. In one embodiment, the Qliveness is a random number generated by the Provider. The Qliveness provides two main security functions 1) makes it harder to reuse Qsid as an attacker would need to know both the Qsid and the Qliveness numbers, and 2) makes it impossible for the Qserver to replay attacks since the Provider can choose the Qliveness. In one embodiment, it is a configuration option if the Provider wishes to create the Qliveness random number or trust the Qserver.

In a second embodiment the user's access software (i.e. browser for web services) also has a plug-in or other technique for validating the user is at the Provider (i.e. web-site) that registered the Qsid. This plug-in could be used to exchange information securely between the Qapp and the user's browser. This might include validation of the URL, session keys, or other information for signing and/or securing communication. For a man-in-the-middle attack where a user has gone to the wrong site (from a spam link, mis-typed the URL, etc) and believe they are at the Provider's web site but instead are at a site run by an attacker, the only assured method to alert the user to their mistake is to

validate at the browser the User's intention with the Qserver. This can be done as a browser plug-in or stand alone program on the desktop.

In different embodiments, some or all of the user credentials stored on the smart phone are stored in an encrypted key-store which includes the private keys and certificates used to communicate with Providers or potentially other users. User credentials could include any information or token to validate a user including public/private key pairs, passwords, biometric templates, one time password seeds, etc.

To decrypt the keystore one embodiment uses a unique split key system to prevent brute force guessing of the decryption key 601 and 606 if the keystore is lost, and prevent the Qserver from having access to the keystore. See FIG 6. It is possible to include the following split key system: where a user enters secret data is split into multiple portions, where one embodiment is to use 2 portions: A and B. The secret data can be any type of user data including: password, pin, biometric template, an image drawn by the user, etc. The A portion of the password is used locally to create the A portion of the decryption key 605. In various embodiments the A portion might be used to decrypt another block on the local system, it might be combined with system information like the smart card's unique serial number, or it might be hashed or otherwise modified to make the A portion longer or more obscure to decode. Since the A portion never leaves the phone the user maintains control over the decryption key. The B portion of the secret data 602 is sent to the Qserver over an encrypted and authenticated channel (using the user's communication key), and the Qserver sends back the B portion of the decryption key 604. In different embodiments the B portion of the key might be looked up from a database, created as a hash from the B portion of the secret data, or combined other user/hardware specific information. Since the Qserver can monitor the use of the B portion it is possible to lock the account after a set number of failed attempts, alert the user, or otherwise respond to suspicious behavior 603.

In one embodiment, the user credentials are based on a public/private key and certificate system (PKI). The benefits of this system are that the user's credentials, the private key, never leave the smart-phone and the

certificates can be signed and validated into the system at enrollment. These certificates can be handed out and controlled individually, allowing users to authenticate with separate credentials to each Provider. In addition other pairs of credentials for example user-to-user credentials can also be created and maintained separately. One embodiment uses three different types of certificates which are signed by three different signing authorities: 1) Provider Certificate Authority (CA) – signs provider certificates for use in communicating between the Provider and the Qserver, 2) User CA – signs the user communication certificates and the user-to-provider certificates created by the user for use in validating their credentials with a Provider. And 3) Qserver CA – signs the certificates used on the Qservers to validate to the user and provider that they are talking to a legitimate Qserver. In a second embodiment, the user might have a single user-to-all-providers certificate, and the signing authorities could be consolidated into a single CA or hierarchical CA configuration.

It is possible for the user to use multiple forms of authentication data including biometrics. The current state-of-the-art in authentication systems combine some form of knowledge, biometric, and token factors; provisioning for just one of these factors is complex without the techniques disclosed herein. Because smart phones and other types of mobile devices have multiple sensors and other data input methods it is possible to collect data from all traditional authentication types:

- Something you are: includes biometric options like face recognition taken from a front facing camera, hand recognition taken from a rear facing camera, speaker recognition taking from the microphone, etc. Some phones have specialized inputs build in like fingerprint readers, and devices can often be attached to the smart phone allowing additional capabilities.
- Something you know: includes passwords, security questions, the phrase spoken during speaker recognition, an image drawn on the screen, a phone shaking pattern, etc.
- Something you have: the most obvious is the phone or mobile device itself. But this category can also include tokens associated with the

phone for example: attached usb devices, tokens discovered via near-field radio options, card data read off of card readers, data from the SIM card of the device, secure data stored on the device itself in a secure co-processor or data lockbox, etc.

5 The techniques disclosed herein allow the Provider and user to use one or more types of authentication giving flexibility and additional security where needed on a individualized basis. The Qserver manages additional factor security including: enrollment of various biometric or token options, storage of security data like biometric templates, fraud detection targeted at the different forms of authentication, and comparing login authentication to the enrolled data. The Qapp manages collecting the authentication data and may do a set
10 of pre-processing steps on the data before submission to the Qserver. The pre-processing is primarily directed at ensuring the quality of the submission to give the user feedback quickly if the image is out-of-focus, not received etc. but the pre-processing is also used to limit the size of data sent and potentially clean up submitted data (align the head to the center of the image, etc.). The Qapp may also have a set of fraud detection functions which are done in
15 addition to the main fraud detection done on the Qserver. Even though the Qserver maintains the additional authentication factors, since the Qserver does not have access to the keystore 406 the Qserver can not pretend to be a user. This separation of trust allows centralized control and management without compromising the user's control over their own authentication. Another benefit of this architecture is new forms of authentication can be
20 rolled out to users without the need for the Provider to change anything.

The architecture as disclosed herein also allows simplified provisioning when Providers want to add new users. There is no need for the Provider to pre-configure the authentication data, for example there is no need to assign temporary passwords. This also allows the Provider to have greater flexibility in the types of authentication policies they want to support. For
25 example if the Provider wants to upgrade the authentication to require 2-factor, they simply change the global policy on the server and users will start being required to authenticate with 2-factors. The Provider does not need to go back to each user trying to collect enrollment data for the new factor that is

all handled by the Authentication Server. When combined with simplified identity and enrollment, there is no need for Providers to pre-configure any account on their systems.

In one embodiment, it would be possible to replace a traditional user with an automated process, like batch system that need to authenticated to a variety of services across a network. Although the system would work the same certain optional components like biometric authentication or knowledge based authentication factors would not be possible.

The techniques disclosed herein may also allow a secure form of communication between the Provider and the User via the secure channel through the Authentication Server and the Qapp on the user's mobile device. This secure communication channel would allow Provider's to send critical notifications in a trusted manner allowing information like "you have made a purchase" or "you have transferred money" to be transmitted in a manner that can not be spoofed or modified. The encryption keys for the data could be based on any standard including the public certificates previously exchanged between the parties or previously shared session keys. This type of communication would also allow secure communication between any two users on the network using a pre-established user-to-user credential.

Unidirectional requests like "you have made a purchase" would not require a response from the user, but bidirectional requests like "are you sure you would like to purchase X" would allows the Provider to wait for a response from the user. Thus allowing the user to validate transactions through the secure communication channel before they are finalized. The verification process could occur even if the user is not currently logged into the Provider and could involve the user "approving" the transaction while simultaneously providing additional authentication data.

Detailed Sample Embodiment of the Login Process-See FIG 3

- User goes to the web site via a desktop or mobile browser 301.
- Web Site 103 (Provider) gets, over channel 104, a Session Id (Qsid) 300 from the Authentication Server 100 and locally creates a session-specific random number (Qliveness) 300. The Provider could pre-cache for performance reasons a set of Qsids from the Qserver. The

Qliveness could be generated on the Authentication Server (with some increased security risk). If the Qliveness codes are generated by the Provider, it will send the Qliveness codes to the Authentication Server as the Qsid is handed out.

- 5 • The Provider shows the Qsid incorporated into the Qcode displayed to the user. The Qcode can be generated by the server and shown as an image or transmitted to the browser and displayed as an image created by the browser side Javascript. The Qsid should be encrypted using SSL or other transport encryption to prevent race conditions if a third
10 party steals the Qsid. In at least some implementations, the user enters or otherwise provides via the browser their unique user ID to the Provider, and initiates the login process. The Provider then sends the unique user ID, the Qsid, and the Qliveness to the Authentication Server over an encrypted channel.
- 15 • In the background, the User's browser continuously polls the Authentication Server over an encrypted channel 105 with the Qsid to identify when the authentication is complete.
- User scans the Qcode with their Authentication Application (Qapp) 302. The user may or may not need to enter their secret data before starting
20 the Authentication Application (based on Qapp policies). The Qapp decodes the Qcode, makes sure it is a proper code and then extracts the Qsid.
- The Qapp connects to the Authentication Server over client-authenticated SSL (so that the Qserver can verify the Qapp user);
25 verifies the Authentication Server's certificate, and then sends the Qsid step 303.
- The Authentication Server sends back to the Qapp the session data 304 (or, if the prior two bullet steps are not performed, the
30 Authentication Server, based on the unique user ID, connects to the user's mobile device and Qapp, over a bi-directional authenticated SSL connection, sending a login request with the session data) which includes:

- The Qliveness number and the type of session (login or enrollment).
- The Distinguished name, the Logo, and a readable name of the Provider the user is attempting to connect to. Optional embodiment would send the Provider's certificate and Qliveness signed with the Provider's key, allowing Provider validation but requiring higher CPU overhead.
- The authentication policy of the Provider required (none, password, biometric, etc.)
- The Qapp checks the user's permissions 408 and 405 (do they want to always be notified, always type their password, etc.) combined with the Provider's authentication policy and then asks the user for the appropriate information 305. For the example, assume the user needs to enter a password and biometric data for three factor authentication.
 - User enters their secret data into the Qapp.
 - Qapp sends the network component of the secret data to the Authentication Server – over the encrypted and validated channel.
 - The Authentication Server – validates the network Component of the secret data, does fraud detection to prevent brute force guessing and other types of attacks, and then sends back the B half of the decryption key for unlocking the keystore on the user's phone.
 - The Qapp receives the B half of the split key, combines it with the A half of the secret data and potentially other information from the phone which is not security relevant (for example the unique ID of the device, etc.) and uses the combined key to unlock the secure keystore in the application. The keystore contains a private key for each Provider the user communicates with.
 - The Qapp then prompts to the user for any additional biometric data required (for example a hand image).
- The Qapp then sends the full authentication packet back to the Qserver 306. Including:
 - The Qsid – The session ID.

- The user id, Qid, is embedded in the communication certificate and can be obtained by the Qserver based on the client-authenticated SSL connection. In different embodiments, the Qid could be unique per user/Provider pair or could be unique per Qapp installation.
- 5 ◦ The Qsid and Qliveness signed by the User's certificate for the specific Provider.
- The raw biometric data (for example the jpg image of their face or hand)
- The Authentication Server then validates the user package including
10 checks for
 - The User Communication Certificate is valid. We make sure the certificate has not be revoked (when a user losses their phone this may happen), is current (certificated will expire and need to be refreshed), and actually exists (new users or attackers would not
15 have a valid certificate).
 - The Qsid is valid. This may include checking for attacks (such as an attacker trying to reuse a Qsid) or brute force scans (such as an attacker sending a random Qsid); and will also include checks to make sure it has not expired, the mapping to a site is valid, etc.
 - 20 ◦ The user has an account with the Provider mapped by the Qsid. A mapping exists from the Qid to the Provider registered with the Qsid. If the user does not have an account, the Authentication Server will send back a "you're not registered with that site message" to the user. This type of check is one of the methods that
25 is used to invalidate phishing attacks.
 - Biometric data is verified against data perviously enrolled by the user. The Biometric data may or may not have a variety of fraud detection steps performed.
 - The signed Qsid and Qliveness are valid and signed by the correct
30 certificate.
- The Authentication Server (which is being polled by the user's original browser) sends back a valid login signal to the user browser.

- The browser connects to the Provider's received authentication approval location.
- The Provider then connects to the Authentication Server (over an encrypted and validated channel) asking for confirmation of the approval 307. The Authentication Server sends back "received a valid login" to the Provider including the following data (Note the preferred method specifically does not send the biometric data or other authentication data to the Provider): the user certificate which includes the Qid, the signed Qsid and Qliveness, the validity of the authentication passed to the Qserver (for example user authenticated successfully with pin and face).
- Optionally, the Provider can then approve or reverify the signature based on local policy and can perform additional security checks on the certificate including matching the login certificate against the enrollment certificate. Once approved, the user is logged into the web site.

Detailed Sample Embodiment of the Enrollment Process

1) User Enrollment to the Qserver

The user downloads the Qapp onto their smart phone. When the Qapp is first started the user can create a new account or enroll into the system:

1. User enters various registration information, which may include name, phone number, e-mail, etc.
2. Optionally, user enters biometric enrollment data like face images, voice prints, password, etc.
3. User clicks "submit" and Qapp sends a request for User ID (Qid) to the Qserver. The request may include various portions of the registration data for example e-mail address to verify the user is not already enrolled and optionally to perform out-of-band validation (like sending the user an e-mail).
4. The Qapp receives back from the Qserver a unique user id (Qid) and the public certificate of the Qserver. The Qapp creates a unique public/private key pair, user communication certificate (which includes the Qid), and a certificate signing request.

5. The Qapp then sends the enrollment data to the Qserver including: the certificate signing request, the registration data, and any biometric data.
6. Assuming the Authentication Service approves the submission, The Authentication Service sends back a signed certificate of the user's communications key and enrolls the user data into the system. This information is then saved in the Qapp.

2a) User Enrollment to Provider

When user connects to a Provider service and chooses to enroll they will be presented with the Provider's existing enrollment page, including any information needed to be supplied by the user, and a Qcode.

- The first part of the enrollment process is the same as the login process. It diverges when the Qapp receives back the session data and it includes the identifier that this is an enrollment session.
- The Qapp shows the user a message to approve enrollment "Provider X is requesting enrollment". If the user approves, the Qapp requests the user to enter any additional authentication required (for example face recognition, etc.).
- The Qapp creates a public/private key pair for use between the User and the Provider.
- The Qapp sends off a certificate signing request.
- The Authentication Server validates parameters and biometrics, as appropriate; and signs the certificate signing request.
- The Authentication Server then sends back to the Qapp: (optional) public certificate for the Provider; and signed certificate for user. The Authentication Server then associates the certificate with the (user, Provider) pair.
- The Qcode image or other visual display on the original browser page will be updated to show successful enrollment. The user can now submit their enrollment to the provider.
- Provider validates session based on session ID and if is successfully validated saves users public certificate and User ID (Qid). If the session ID is not valid it could just be a user enrolled on the Provider

site without using the authentication server, so the session ID was never used.

2b) Secondary Enrollment Embodiment – Addition of Identity

One way to make user enrollment at a Provider site easier on the user is to simplify the amount of information the user has to re-enter. See FIG 8. The techniques disclosed herein may allow the user to create one or more “Identities”, for example business and personnel identities 806. Each Identity has a set of data for example name, e-mail address, mailing address, etc. that is associated with the Identity. The user during enrollment then has the option of using information from an Identity to fill out Provider enrollment information. This simplified enrollment option could be triggered when a user scans a login Qcode or enters their unique user ID into the Provider's form, and the Qserver recognizes they do not have an account. In one embodiment, the identity information can be stored on the Qserver and in a second embodiment the identity could be stored exclusively on the mobile user device. Additionally, the management of the identity information could be performed locally to the storage of the information or done with remote agreement on another server or device: including a server, the authentication server, the users desktop computer, etc.

- The first part of the enrollment process is the same as the login process. It diverges when the Qapp receives back the session data and it includes the identifier that this is an enrollment session.
- The Qapp shows the user a message to approve enrollment “You do not have a login for Provider X would you like to enroll?”. The type of enrollment information required by the Provider (name, date of birth, etc.) is sent as a set of properties to the Qapp by the Qserver. The Qapp then shows the user a message like “To enroll Provider X would like the following information: name, email, etc.”. The user can then select from their set of Identities which one they would like to use for the Provider and the required fields would be filled in using the pre-configured Identity data. Optionally the user may be given the choice to

- edit the data before being submitted. The user maintains complete control over the data submitted and yet can do a click to enroll process – potentially never having to type in any new data.
- The enrollment process then proceeds normally.

5 3) Provider Enrollment to the Qserver

This is expected to happen much less frequently than user enrollment and as such the signing key and process can be more manual. It essentially follows the same steps above except the Provider creates their key (using provided scripts) and saves the data that is returned as part of their provider configuration. The Provider receives back the Authentication CA and the User CA certificates.

Detailed Sample Embodiment of the Lost Phone Process

See FIG 7. Because users certificates are validated before login, by using the proposed system, when a user reports a phone lost 705; all the certificates can be immediately invalidated. This can also be used if fraud is detected or the user thinks their phone may have been compromised. To help assist the user not only are the keys revoked but the Qserver service can be used to simplify and manage re-enrollment with their old providers (in fact it is not necessary for the providers to do anything if they trust the Qserver to revalidate the user). In different embodiments, the methods of invalidating the credentials include key revocation, deletion, or invalidating the data.

The user reports their phone lost by calling or logging into the Qserver web site. They can use the B half of the secret data or if they have a new phone login with biometric options.

- 25 • The Authentication Service revokes (or marks as revoked all the user's keys).
- If a user tries to login with an revoked key the login is denied.

Revalidation

- 30 • When a user goes to reset up their account, on the Qapp they select "Login to Existing Account" and give the details for login including e-mail address, biometrics, and network portion of the secret data. The appropriate information is sent onto the Authentication Server for

validation to ensure that the biometrics and other login information is correct.

- If the authentication is valid then the Authentication Server sends back the user's old User ID (Qid). The normal enrollment continues with the Qapp creating a key pair, certificate, etc.
- Then when a user tries to login to a site they used to have credentials on, they follow the normal login process except that the Authentication Server sees that no current credentials are found for the user, yet they have revoked credentials, and contacts the Qapp to create new credentials. Once the credentials are created the user's original browser is contacted and told the user is doing a re-enrollment. By using a browser return code to notify the browser the redirection can happen automatically to the user. The ability to give the user a completely transparent re-enrollment option is enabled by the authentication framework, and the specialized communication between the browser and the provider. This gives the provider the opportunity to redirect the user to a re-enrollment page where the provider can ask additional questions to re-verify the user (for example Favorite Pet's Name, etc.). The provider can also choose to skip this step and just accept the new credentials. Once the credentials are accepted the user logs in. The Authentication Server is also contacted by the provider to "accept" the new credentials.

Detailed Sample Embodiment of the Policy Configuration

- There are a number of policy options and configurations that the Provider and User can select. Each effects slightly the steps taken to login or validate a login process. For example the Provider can specify that the user's key should be stored in an encrypted key store or that the user is required to use two-factors to login. The User can also specify if keys should be stored encrypted. The Qapp selects the minimal settings that meet both the User and Provider specifications. This means that each of the user related options acts as an "upgrade" to the security. The policy management can be distributed for example the Qapp or the Qserver can be used to change user policies. Where as the Qserver or the Provider might have access to change provider policies.

In one embodiment, only the provider can update provider policies. In one embodiment only the Qapp can change the user policies. In another embodiment, the user could change their policies from an interface, like a web site, on the Qserver. User's may be given control over their individual policy information and records, rather than having an administrator or super-user who is responsible for maintaining a multitude of individuals records.

User Policies:

- Encrypt all keys. This will require the user to enter their secret data, whenever the Qapp starts on the system.
- Allow authentication to remain valid (i.e. stored in memory) for a set period of time. This allows the user to limit the number of times they enter their secret data or take an image of their, possible settings include: every time the screen saves, or maybe every hour. Specific biometrics or speciality tokens may have their own maximum time frames to remain valid.
- The User can directly manage (either on the Qapp application or on the Qserver web site) the policies for specific keys 801. This would include "upgrading" specific sites to require more authentication. For example, if the provider currently requires two-factor (the phone and your secret data), the user can upgrade the requirement to add a biometric factor so that now for that User their account on the specific provider can not be accessed without providing three-factor authentication.
- In another embodiment the user could select to receive Provider certificates and Qliveness signature blocks and validate the signatures at the Qapp.

Provider Policies

Configuration options may be included which allow the provider to trust the Qserver and skip most of the provider checks or allows the Provider to revalidate everything from the user (except the additional authentication factor(s)). The following are some of the major settings which can be used on the Provider:

- Revalidate the user credentials. This includes: validating that the user credentials are the same as was approved during enrollment; the certificate has not expired; and the signed data returned during authentication was signed using the previously agreed certificate.
- 5 • Creating their own Qliveness rather than having the Qserver create it when the Qsid is obtained. (This prevents replay attacks being run by a Qserver).
- Turning on or off re-enrollment. This allows the Provider to ask the user re-verification questions if they lose their key and have to be re-
- 10 enrolled. If turned off the Provider trusts that the Qserver has done the authentication verification.
- Use a cookie given to the user's browser when they first connect to the Provider to validate that the browser that saw the Qcode is the same one that logged in.

15 4) Existing User Enrollment to Provider with Paper

One benefit of using a Qsid transmitted to the user's mobile device through a QR code is that the initial enrollment for existing Provider accounts can be done via a mailer or paper initiation. The benefits of this are that the enrollment process could be initiated via a Provider statement (like an account statement or utility bill) or during initial setup (like when you go to open a bank

20 account or get a home loan). See FIG 10. Under this enrollment process the following steps would occur:

- When the provider is setting up the account or wishes to enroll an existing user they print on a piece of paper 1000 a Qcode 1008 that is
- 25 unique allowing the user and Provider account to be correlated through the Authentication Server. The Qcode includes a header block (described below) and an Enrollment Id (QEid) agreed upon by the server 1001, which is guaranteed to be unique across all users for as long as the enrollment code is valid and acts as a simple identifier for
- 30 the enrollment session.
- User scans 1002 the Qcode with their Authentication Application (Qapp) 101. The user may or may not need to enter their secret data before starting the Authentication Application (based on Qapp policies).

The Qapp decodes the Qcode, makes sure it is a proper code and then extracts the Qsid.

- The Qapp connects to the Authentication Server over client-authenticated SSL (so that the Qserver can verify the Qapp user);
5 verifies the Authentication Server's certificate, and then sends the Qeid 1003.
- The Authentication Server sends back to the Qapp the session data 1004 which includes:
 - The Qliveness number and the type of session (login or enrollment).
 - 10 ◦ The Distinguished name, the Logo, and a readable name of the Provider the user is attempting to connect to. Optional embodiment would send the Provider's certificate and Qliveness signed with the Provider's key, allowing Provider validation but requiring higher CPU overhead.
 - 15 ◦ The authentication policy of the Provider required (none, password, biometric, etc.)
- The Qapp shows the user a message to approve enrollment "Provider X is requesting enrollment". If the user approves, the Qapp requests the user to enter any additional authentication required 1005 (for
20 example face recognition, etc.).
- The Qapp creates a public/private key pair for use between the User and the Provider.
- The Qapp sends off a certificate signing request and other optional authentication information 1006.
- 25 • The Authentication Server validates parameters and biometrics, as appropriate; and signs the certificate signing request.
- The Authentication Server then sends back to the Qapp: (optional) public certificate for the Provider; and signed certificate for user. The Authentication Server then associates the certificate with the (user,
30 Provider) pair.
- The Authentication Server then sends the enrollment information including newly generated user-provider certificate to the Provider 1007. The Provider may be contacted through any number of

architectures including the Provider continuously polls the Qserver, they are connected continuously, or the Qserver has the ability to directly connect to the Provider. The Provider validates the session based on the enrollment ID and if is successfully validated saves users public certificate and User Account ID (Qid).

The user is then enrolled into the authentication system, without typing any additional validation information, and can now login with simplified multi-factor authentications. In a second embodiment the Provider may request additional information be entered by the user on first use – similar to the re-enrollment process there by providing another layer of validation.

Detailed Sample Embodiment of the Notification Process

1) Provider Sends a Message to a User

Once the user-provider relationship has been configured (through the enrollment process) the Provider has a unique User Account ID associated with the specific user account. See FIG 11. If an event occurs at the Provider which should be validated or messaged to the user the following process would occur:

- The Provider send to the Qserver the User's Account ID (Qid), a transaction ID, and a message, which can be encrypted with the user's certificate already stored at the Provider and signed by the Provider certificate, message 1100. The request would also include a header specifying if the user needs to approve the transaction, and if any authentication is required for approval. The transaction ID will be used to later identify the specific transaction to verify delivery or approval. In a second embodiment the Qsid of a already logged in user could be used instead of the Qid to correctly identify the user for the message.
- The Qserver initiates a connection to the user mobile device Qapp 1101. The connection initiation could occur over a push mechanism, an open network port, initiated via SMS, or the Qapp could poll or stay connected to the Qserver regularly depending on the mobile device architecture and services available.
- The Qserver sends the Qapp a message request including the Transaction ID, message block, and policy requirements. The Qapp

then decrypts the message, shows the message to the user, if required obtains approval and additional authentication information for validation 1102.

- The Qapp then generates a response and sends it back to the Qserver 5 1103 which forwards it back to the Provider 1104. The response may include the user's approval answer, additional authentication information, and verification that the message was shown. In one embodiment the Provider may poll the Qserver continuously for responses to one or more messages. In a second embodiment the 10 user's browser would poll the Qserver after the user initiated a transaction that required verification, and when completed the browser tells the Provider to check the status of the notification. In a third embodiment the Qserver directly connects to the Provider.

This same process could be used to share messages between two 15 users that have accounts on the Qserver and have already exchanged certificates.

CLAIMS

We claim:

1. A method of authenticating a network user to another network entity, comprising:
 - executing, on a first user operated device, a first program to:
 - receive user inputted validation information;
 - store a user credential on the first user operated device;
 - executing, on a second user operated device, a second program to:
 - receive information from another network entity via the network;
 - further executing the first program to:
 - receive an input transferring, to the first program, the information received by the second program from the other network entity;
 - direct transmission, to the authentication server via the network, of the transferred information;
 - receive, from the authentication server via the network, an identifier of the other network entity, other information, and authentication policy requirements of the other network entity;
 - direct transmission, to the authentication server via the network, of the input validation information corresponding to the received other network entity authentication policy requirements;
 - receive, from the authentication server via the network after directing transmission of the validation information, a request for a user credential;
 - sign a message, including the transferred information and the received other information, with the stored user credential; and
 - direct transmission, to the authentication server via the network, of the signed message to authenticate the user; and
 - further executing the second program to:
 - receive, from at least one of the authentication server and the other network entity via the network, an indication that the user has been successfully authenticated.

2. The method of claim 1, wherein:
 - the first user operated device and the second user operated device are the same device;
 - the information is a random number that serves as a session identifier; and
 - the other information is another random number.

3. The method of claim 1, further comprising further executing the first program to:

generate a private/public key pair for the user, wherein the stored user credential is the private key of the generated user private/public key pair; and
direct transmission, to the authentication server via the network, of the public key of the generated user private/public key pair.

4. The method of claim 1, further comprising further executing the first program to:

generate user secret data;
divide the generated secret data into multiple portions including a first portion and a second portion;
encrypt the user credential with the generated secret data, wherein the stored credential is the encrypted credential;
direct transmission, to the authentication server via the network, of the second portion of secret data;
also receive, from the authentication server via the network after directing transmission of the validation information, the second portion of secret data;
combine the stored first portion of secret data with the received second portion of secret data;
and
decrypt the stored encrypted credential with the combined portions of secret data;
wherein the message is signed with the decrypted user credential.

5. The method of claim 1, further comprising further executing the first program to:

receive user inputted user authentication policy requirements;
direct transmission of the received user authentication policy requirements to the authentication server;
also receive from the authentication server with the authentication policy requirements of the other network entity, any additional authentication policy requirements based on any differences between the user authentication policy requirements and the other network entity authentication policy requirements; and
also direct transmission, to the authentication server via the network, of the received validation information corresponding to any received additional authentication policy requirements.

6. The method of claim 1, wherein the information is first information and the other information is first

other information, and further comprising, after the authentication server has been notified that the first user operated device is no longer in use or has been lost or stolen or the user credential has otherwise been compromised:

 further executing the second program to:

 receive, from the other network entity via the network,
second information;

 executing, on the first or a third user operated device, the first program to:

 receive an input transferring, to the first program, the
 second information received by the second program from the other
network entity;

 direct transmission, to the authentication server via the
network, of the transferred second information;

 receive, from the authentication server via the network, an identifier of the
other network entity, second other information, and authentication policy
requirements of the other network entity;

 direct transmission, to the authentication server via the network, of the
input validation information corresponding to the received other network entity
authentication policy requirements; and

 receive, from the authentication server via the network in response to the
transmitted validation information, a notification that the user has been validated
but cannot be authenticated because the user credential has been invalidated.

7. The method of claim 6, further comprising further executing the second program to receive, from the authentication server via the network, a redirection instruction, redirecting the user to the other network entity's reenrollment website on the network.

8. The method of claim 6, further comprising further executing the first program to:

 receive a request for a replacement credential from the authentication server via the network;
and

 in response to the received request for replacement credential, generate a replacement
credential, store the generated replacement credential, and direct transmission of the generated
replacement credential to the authentication server via the network.

9. The method of claim 6, further comprising, after directing transmission of the generated

replacement credential to the authentication server, further executing the first program to:

direct transmission, to the authentication server via the network, of another message, including the second information and the second other information, signed with the replacement credential to authenticate the user.

10. The method of claim 1, further comprising further executing the second program to:

direct a presentation, on a display screen, of the received information;

wherein the received information is in the form of an optical code; and

wherein the input transferring the received information to the first program, is a digital photograph of the presented optical code.

11. The method of claim 1, further comprising further executing the second program to:

direct printing of the received information;

wherein the input transferring the received information to the first program is a digital photograph of the printed optical code.

12. An article of manufacture for authenticating a network user to another network entity, comprising:

non-transitory storage medium; and

logic stored on the storage medium, wherein the stored logic is configured to be readable by a processor and thereby cause the processor to operate so as to:

receive user inputted validation information;

store a user credential;

receive an input of information, wherein the information was obtained by the user from another network entity;

direct transmission, to the authentication server via the network, of the input information;

receive, from the authentication server via the network after directing transmission of the information, an identifier of the other network entity, other information, and authentication policy requirements of the other network entity;

direct transmission, to the authentication server via the network, of the input validation information corresponding to the received other network entity authentication policy requirements;

receive, from the authentication server via the network after directing transmission of the validation information, a request for a user credential;

sign a message, including the transferred information obtained from the other network entity and the other information received from the authentication server, with the stored user credential; and

direct transmission, to the authentication server via the network, of the signed message to authenticate the user.

13. The article of manufacture of claim 12, wherein the stored logic is further configured to cause the processor to operate so as to:

- generate a private/public key pair for the user, wherein the stored user credential is the private key of the generated user private/public key pair; and

- direct transmission, to the authentication server via the network, of the public key of the generated user private/public key pair.

14. The article of manufacture of claim 12, wherein the stored logic is further configured to cause the processor to operate so as to:

- generate user secret data;

- divide the generated secret data into multiple portions including a first portion and a second portion;

- encrypt the user credential with the secret data, wherein the stored user credential is the encrypted credential;

- direct transmission, to the authentication server via the network, of the second portion of secret data;

- also receive, from the authentication server via the network after directing transmission of the validation information, the second portion of secret data;

- combine the first portion of secret data with the received second portion of secret data; and

- decrypt the stored encrypted credential with the combined portions of secret data, wherein the signed message is signed with the decrypted user credential.

15. The article of manufacture of claim 12, wherein the stored logic is further configured to cause the processor to operate so as to:

- receive user inputted user authentication policy requirements;

- direct transmission of the received user authentication policy requirements to the authentication server via the network;

- receive, from the authentication server via the network with the authentication policy requirements of the other network entity, any additional authentication policy requirements based on any differences between the user authentication policy requirements and the other network entity authentication policy requirements; and

also direct transmission, to the authentication server via the network, of the input validation information corresponding to any received additional authentication policy requirements.

16. The article of manufacture of claim 12, wherein, the information is first information and the other information is first other information and, after the authentication server has been notified that the user credential has been compromised, the stored logic is further configured to cause the processor to operate so as to:

- receive an input second information, wherein the second information was obtained by the user from the other network entity;

- direct transmission, to the authentication server via the network, of the input second information;

- receive, from the authentication server via the network after directing transmission of the second information, an identifier of the other network entity, second other information, and authentication policy requirements of the other network entity;

- direct transmission, to the authentication server via the network, of input validation information corresponding to the received other network entity authentication policy requirements; and

- receive, from the authentication server via the network after directing transmission of the validation information, a notification that the user has been validated but cannot be authenticated because the user credential has been invalidated.

17. The article of manufacture of claim 16, wherein the stored logic is further configured to cause the processor to operate so as to:

- receive a request for a replacement credential from the authentication server via the network;

- in response to the received request for the replacement credential, generate a replacement credential, store the generated replacement credential, and direct transmission of the generated replacement credential to the authentication server via the network; and

- direct transmission, to the authentication server via the network, of another message, including the input second information from the other network entity and the second other information from the authentication server, signed with the replacement credential to authenticate the user.