

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

According to certain aspects of the invention any of multiple different users can be authenticated to any of multiple different network service providers via the network, e.g. the Internet.

The system includes an authentication server, which is sometimes referred to as the Qserver, and could be in the form of a cluster of servers. The authentication server is configured with logic, e.g. programmed with software, to store a provider identifier for each of the multiple different service providers, e.g. banks, brokers, merchants, etc., in association with provider authentication policy requirements for the applicable service provider. The provider authentication policy requirements could require that one or more factors be used to authenticate a user, where the first factor is a credential and other factors are validation information. Any factor(s) could, for example, include a password, other knowledge based data, such as a date of birth, a token and/or biometric data. The authentication server also stores one or more user identifier for each of the multiple different users in association with validation information for the application user.

A first device, such a personal, laptop or tablet computer or smart mobile communications device, such as a smart phone, is operable by a first of the multiple different users and configured with logic to transmit a login request to a first of the multiple different service providers via the network. A network server associated with the first service provider is configured with logic to receive the login request and to transmit a request for a random number and optionally transmit other information, which could be another random number and is sometimes characterized as a liveness id, to the authentication server via the network. The authentication server is further

public key of a private/public key pair of the first user to the authentication server via the network. The private key of the first user private/public key pair is known only to the second device. The authentication server is further configured to store the transmitted first user public key, and transmit a
5 certificate, including the first user public key, signed with the authentication server private key to the second device via the network. The public key of the authentication server private/public key pair is known to the second device.

It should be understood that the above described aspects of the invention can be implemented using logic, for example in the form of software,
10 such as a programmed application or app, stored on non-transitory storage medium, such as a hard, compact or other type disk, or memory, which is configured to be readable by a processor and thereby cause the processor to operate so as to perform the functions and/or operations described above. It should also be understood that the above described aspects of the invention
15 can be implemented on a network server, including multiple servers in a cluster, or smart communications device, as applicable, having a processor configured with the logic to perform the functions and/or operations described above and a data store, e.g. disk or memory, configured to store data as directed by the processor.

20 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
25 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.

30 **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.

5 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.

10 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.

15 **DETAILED DESCRIPTION**

OVERVIEW

First Authentication Framework

According to aspects of the invention provides 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 (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.

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
5 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
10 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.

Authentication Server

15 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
20 700 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
25 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

30 According to aspects of the invention, an authentication framework includes 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

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
5 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
10 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
15 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
20 limiting the scope to the embodiments disclosed herein.

The present invention is 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):

- 25 • 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
30 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

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
5 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
10 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
15 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.
20 See FIG 7. For example one strength of the present invention 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
25 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
30 gain any additional access to a Provider. Further, because the authentication 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

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.

- 5 1. The user goes to a web site that implements the present invention's 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
10 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
15 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.
- 20 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
25 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
30 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.

5 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 present invention's login system. Instead of the traditional username and password fields the web site has a single field for the user to enter their
10 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
15 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.
- 20 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
25 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.
- 30 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 for the present invention, 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.

10 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 15 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 20 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 25 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.

30 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. The present invention includes 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)

- 5 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.

10 In the present invention 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 present invention. Because smart phones and other types of mobile devices
15 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.
20 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
25 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
30 SIM card of the device, secure data stored on the device itself in a secure co-processor or data lockbox, etc.

The present invention allows 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

5 Qapp manages collecting the authentication data and may do a set 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

10 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 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

15 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 rolled out to users without the need for the Provider to change anything.

The architecture of the present invention also allows simplified

20 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 example if the Provider wants to upgrade the authentication to require 2-

25 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

30 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 present invention also allows 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 allow 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.
- 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

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.

- 5 • 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.).
- 10 • 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.
- 15 • 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.
- 20 • 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
- 25 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

30 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 present invention allows 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

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. In the present invention user's are 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

The present invention includes configuration options 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.
- 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-enrolled. If turned off the Provider trusts that the Qserver has done the authentication verification.

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 certain examples, any of multiple different users can be authenticated to any of multiple different network service providers via the network, e.g. the Internet.

An exemplary system includes an authentication server, which is sometimes referred to as the Qserver, and could be in the form of a cluster of servers. The authentication server is configured with logic, e.g. programmed with software, to store a provider identifier for each of the multiple different service providers, e.g. banks, brokers, merchants, etc., in association with provider authentication policy requirements for the applicable service provider. The provider authentication policy requirements could require that one or more factors be used to authenticate a user, where the first factor is a credential and other factors are validation information. Any factor(s) could, for example, include a password, other knowledge based data, such as a date of birth, a token and/or biometric data. The authentication server also stores one or more user identifier for each of the multiple different users in association with validation information for the application user.

A first device, such a personal, laptop or tablet computer or smart mobile communications device, such as a smart phone, is operable by a first of the multiple different users and configured with logic to transmit a login request to a first of the multiple different service providers via the network. A network server associated with the first service provider is configured with logic to receive the login request and to transmit a request for a random number and optionally transmit other information, which could be another random number and is sometimes characterized as a liveness id, to the authentication server via the network. The authentication server is further

public key of a private/public key pair of the first user to the authentication server via the network. The private key of the first user private/public key pair is known only to the second device. The authentication server is further configured to store the transmitted first user public key, and transmit a
5 certificate, including the first user public key, signed with the authentication server private key to the second device via the network. The public key of the authentication server private/public key pair is known to the second device.

It should be understood that the above described techniques can be implemented using logic, for example in the form of software, such as a
10 programmed application or app, stored on non-transitory storage medium, such as a hard, compact or other type disk, or memory, which is configured to be readable by a processor and thereby cause the processor to operate so as to perform the functions and/or operations described above. It should also be understood that the above described techniques can be implemented on a
15 network server, including multiple servers in a cluster, or smart communications device, as applicable, having a processor configured with the logic to perform the functions and/or operations described above and a data store, e.g. disk or memory, configured to store data as directed by the processor.

20 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
25 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.

30 **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.

5 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
10 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.

15 **DETAILED DESCRIPTION**

OVERVIEW

First Authentication Framework

According to an example, there is provided an authentication framework, see FIG 1, that includes a mobile device of the user which has a
20 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
25 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
30 by the trusted component, may include the network service's address or URL (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.

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

5 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

10 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.

Authentication Server

15 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

20 700 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

25 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

30 According to an example, an authentication framework includes 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

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
 5 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
 10 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
 15 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
 20 limiting the scope to the embodiments disclosed herein.

The techniques disclosed herein are 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):

- 25 • 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
 30 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

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
5 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
10 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
15 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.
20 See FIG 7. For example one strength of the present example 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
25 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
30 gain any additional access to a Provider. Further, because the authentication 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

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.

- 5 1. The user goes to a web site that implements the present example's 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
 10 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
 15 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
 20 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
 25 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
 30 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.

5 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 present example's 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
10 902. 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.
15 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
20 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
25 905. 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,
30 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 for the present example, 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. The present example includes 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)

- 5 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.

10 In the present example, 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 present example. Because smart phones and other types of mobile devices
15 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.
20 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
25 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
30 SIM card of the device, secure data stored on the device itself in a secure co-processor or data lockbox, etc.

The present example allows 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

5 Qapp manages collecting the authentication data and may do a set 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

10 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 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

15 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 rolled out to users without the need for the Provider to change anything.

The architecture of the present example also allows simplified

20 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 example if the Provider wants to upgrade the authentication to require 2-

25 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

30 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 present example also allows 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 allow 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.
- 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

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 FIIG 8. The present example allows 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

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. In the present example, user's are 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.

- 5 • 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.
- 10 • 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.
- 15 • In another embodiment the user could select to receive Provider certificates and Qliveness signature blocks and validate the signatures at the Qapp.

Provider Policies

- The present example includes configuration options which allow the provider to trust the Qserver and skip most of the provider checks or allows the
- 20 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:
- 25 • 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.
 - 30 • 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-enrolled. If turned off the Provider trusts that the Qserver has done the authentication verification.