

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 aspects for the invention, 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

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 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 aspects of the invention, 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 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 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 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 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 aspects of the invention, 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

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
5 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.

10 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
15 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
20 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
25 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, in accordance with the invention, the authentication server can serve as both an authentication server and a certificate authority, and the
30 first user device obtains both user certificates and user authentications from the same entity, i.e. the authentication server.

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

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

- 5 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
- 10 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
- 15 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.
- 20 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

- 25 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
- 30 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

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

Fourth Authentication Framework

According to aspects of the invention, an authentication framework
5 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
server, is also included. Access software is used to reach the network
10 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 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
15 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.

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
20 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 knowledge based factors. The biometric data
25 could be collected from devices attached to the user mobile device

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
30 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

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.

5 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

10 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
15 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.

20 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
25 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 present invention is based on a multi-party system (the user, the authentication service provider 100, and the network service provider 103)
30 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

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.

- 5 • 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
10 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
15 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
20 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.
25 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 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
30 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)

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 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 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 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 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 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.

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
5 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
10 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
15 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,
20 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
25 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
30 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.

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 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 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 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.

10 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 addition to the main fraud detection done on the Qserver. Even though the Qserver maintains the additional authentication factors, since the Qserver

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

25 The architecture of the present invention 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

30 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.

5 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.

10 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
15 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.

20 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
25 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

- 30
- 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

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 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 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. Additional, 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

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
5 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:

- 10 • 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
15 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,
20 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.
- 25 • 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
30 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:

CLAIMS

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.

18. A method of operating a user network device to enroll a network user with another network entity, comprising:

- receiving a user inputted set of user identities, user inputted enrollment data, and a user selection of the particular inputted enrollment data to be associated with each respective identity in the received set of identities;

- storing each respective identity in association with the particular enrollment data selected to be associated with that identity;

- receiving an input of information obtained by the user from another network entity in response to an enrollment request;

- transmitting the input information to the authentication server via the network;

- receiving an identifier of the other network entity and enrollment data requirements of the other network entity from the authentication server via the network;

- receiving a user input selecting one of the stored user identities;

in response to receipt of the inputted selected identify, automatically retrieving the particular enrollment data stored in association with the selected user identity; and

transmitting the retrieved enrollment data to the authentication server via the network to enroll the user with the other network entity.

19. The method of claim 18, further comprising:

transmitting, to the authentication server via the network, a new credential for use with the other network entity and a request that the authentication server sign the transmitted new certificate with a private key of a private/public key pair of the authentication server to create a certificate, wherein the public key of the authentication server private/public key pair is known to the other network entity; and

receiving the certificate from the authentication server via the network.

20. A method of operating a user network device to notify a user of a transaction with another network entity via a network, comprising:

receiving, from an authentication server via the network, a transaction identifier, transaction approval and authentication requirements, and a message regarding the transaction, wherein the message is (i) encrypted with a public key of a private/public key pair of the user, wherein a private key of the user private/public key pair is known only to the user, and (ii) also signed with a private key of a private/public key pair of the other network entity, wherein a public key of the other network entity private/public key pair is known to the user;

validating and decrypting the message by applying the public key of the other network entity private/public key pair and the private key of the user private/public key pair to the signed encrypted message;

directing presentation of the validated decrypted message to the user;

receiving, based on the received transaction approval and authentication requirements, a user input representing a transaction approval, or authentication information, or both; and

transmitting, to the authentication server via the network based on the received user input, at least one of a transaction approval and authentication information.

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

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 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 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 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 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 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

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

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

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

Fourth Authentication Framework

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 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.

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 knowledge based factors. The biometric data could be collected from devices attached to the user mobile device

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

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

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 loses 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)

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.

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 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 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 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 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

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. Additional, 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

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:

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.