wwpass®

How WWPass Authentication Works

vs.

We each have dozens of bank and loyalty cards, an overweight keychain and numerous online accounts. Every time we need to authenticate ourselves, either physically or online, we are forced to provide a card or a username and password. Every day we must choose between a convenient but insecure single universal password, or an inconvenient but secure approach of hundreds of unique, strong passwords that are impossible to remember. At the same time, the organizations with which we authenticate collect our personal information. They may keep the information private, use it for marketing purposes, sell it or allow it to be compromised. We have no control over how our private data is used.

WWPass provides an answer to these security threats with convenient, secure authentication and data storage.

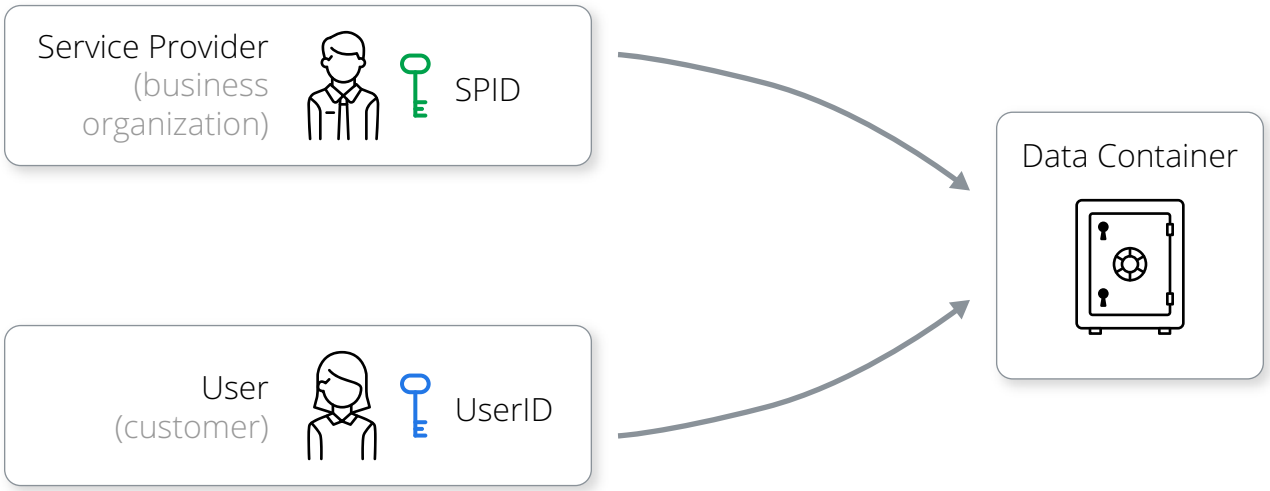
Here's how it works.

WWPass participants

There are three parties in a WWPass model:

1. Service provider with an application or web service, such as websites, cloud services, VPNs or remote desktops.
2. User with a user terminal, which is a home or office computer, retail store checkout, vending machine, etc. To start a WWPass transaction, the user will connect their PassKey to the user terminal.
3. WWPass core network, providing a mutual authentication service between all participants and storing the user's private data in a secure dispersed data container (which is encrypted, fragmented and dispersed throughout WWPass data centers).

Safe deposit box: two-key approach (directed identity)



The core concept in a WWPass solution is the data container, which acts like a digital safe deposit box. A safe deposit box can only be opened with the customer's key, the bank's own guard key, the proper signature and a code.

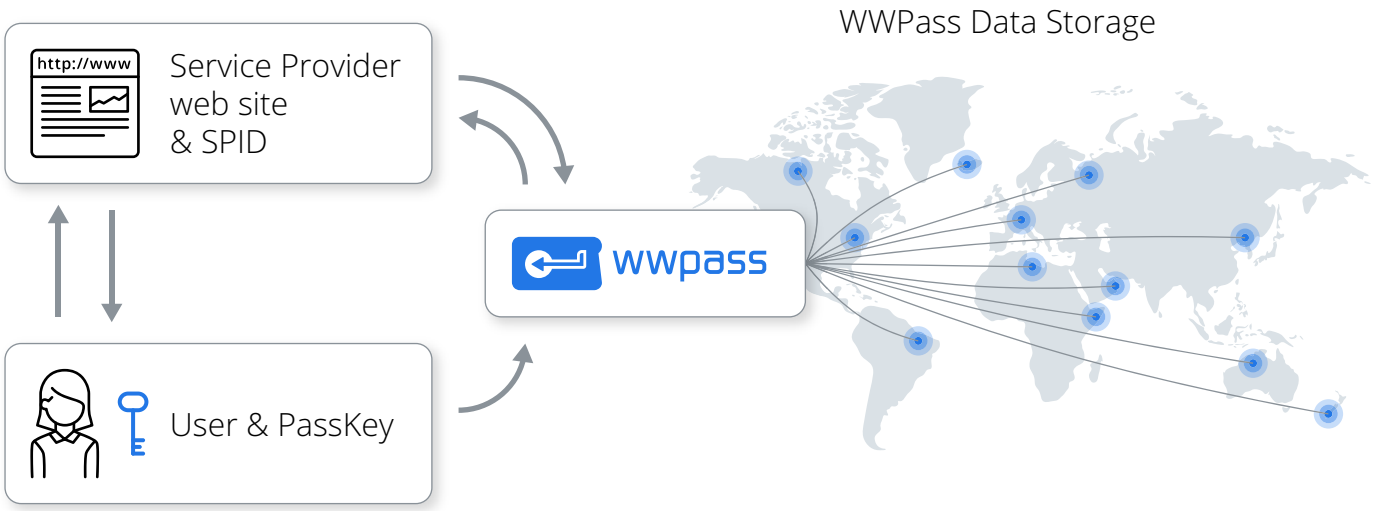
WWPass provides a unique key to every user (the UserID) and a unique key to the service provider (the SPID - service provider ID). The service provider is a network application and may be a mail server, an online retail shop, a corporate web page, a vending machine, a bank or any other application requiring authentication.

WWPass provides a unique data container (analogous to the safe deposit box) which corresponds to each user registration at a particular service provider's website or application (i.e. for each UserID/SPID pair). To open this container, two keys are needed: the UserID key and SPID key. This way users' data belonging to different service providers are fully isolated from each other and personal information that the user chooses to give to one service provider will never be given to any other service provider without explicit user permission.

For an even greater level of security, service providers can implement two-factor authentication and ask users to provide a PIN (analogous to the PIN of smart cards) in addition to their key.

Here's what a WWPass-enabled login looks like:

1. The user requests to log in to a service provider's website or application
2. The service provider communicates with the WWPass data center and provides its key (SPID)
3. The service provider asks the user to provide a key (UserID) to WWPass
4. WWPass uses both keys to open the corresponding data container and passes the contents to the service provider



PassKeys

Just like physical keys, digital keys may be copied and users may not notice for some time. This is why WWPass puts enormous effort into preventing unauthorized key copies. WWPass stores the UserID and other secret data in a secure hardware device called a PassKey.

These PassKeys are Java Cards based on the same smart card technology, which is used in chip cards and mobile phone SIMs. One PassKey can replace most existing cards, keys and username-password pairs for users authenticating to WWPass-enabled websites or applications – a much more convenient and secure option.

Java Cards are able to perform most of crypto operations – symmetric and asymmetric cipher, digital signature, message integrity check, etc. – and can easily be adapted to most kinds of protocols or technologies. Java Cards are cryptographically strong, making it impossible for a hacker to read the contents of the cards to access crypto keys or other secrets.

Java Cards are an industry standard, which makes WWPass PassKeys inexpensive and robust while providing the highest level of data protection. PassKeys are available in multiple form factors and interfaces, including bare plastic cards, NFC tokens, USB keys and others.

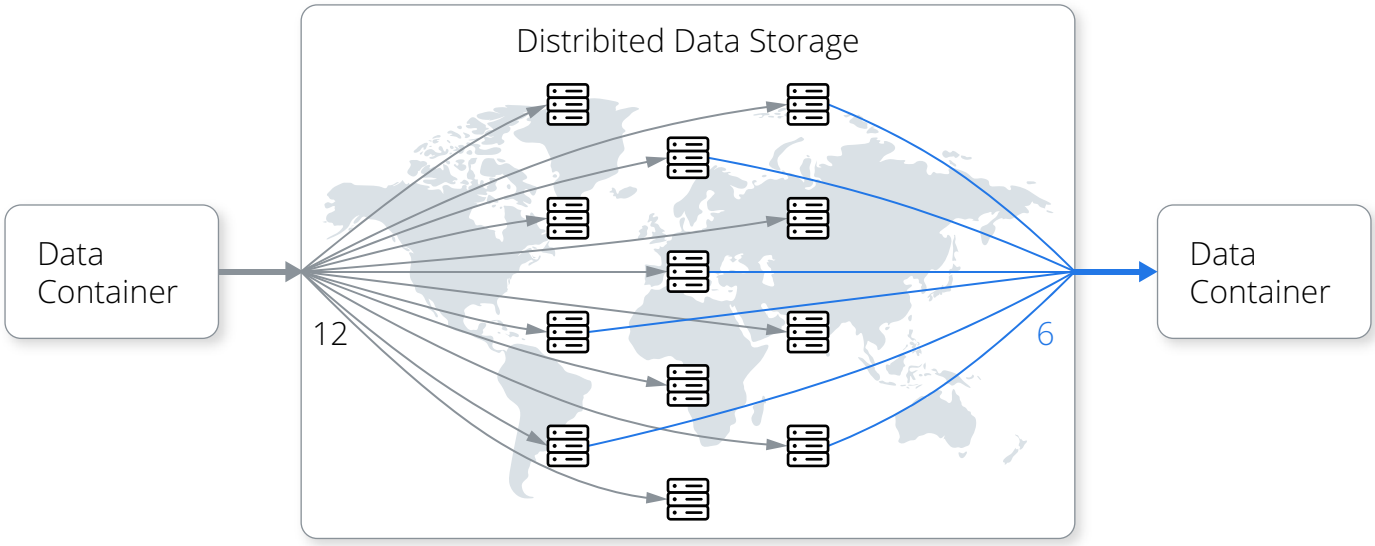
PassKey Lite

Instead of physical PassKeys, users may authenticate to WWPass enabled websites and applications with their smart phones. WWPass PassKey Lite is a mobile application for iOS and Android platforms that turns a smartphone or tablet into a lightweight and easy-to-use authentication device. To authenticate with PassKey Lite a user needs to launch PassKey Lite app on their mobile device and scan the QR-code displayed on the authentication screen of the application or website. PassKey Lite will ask the user to confirm authentication and may ask to enter their PIN as a second factor. Alternatively, if the user needs to authenticate on a website opened on a mobile device, they just tap the authentication QR-code with their finger. This will launch PassKey Lite and start authentication. PassKey Lite will return to the website when authentication is completed.

Digital world: encryption & data dispersion

WWPass encrypts all data, making it unreadable, and takes multiple steps to further secure digital data:

- Every data container is ciphered with its individual cipher key. There is no master cipher key, which allows to decode all WWPass information.
- All data is dispersed. WWPass distributes data content redundantly over multiple geographic locations, in order to survive storage node faults.

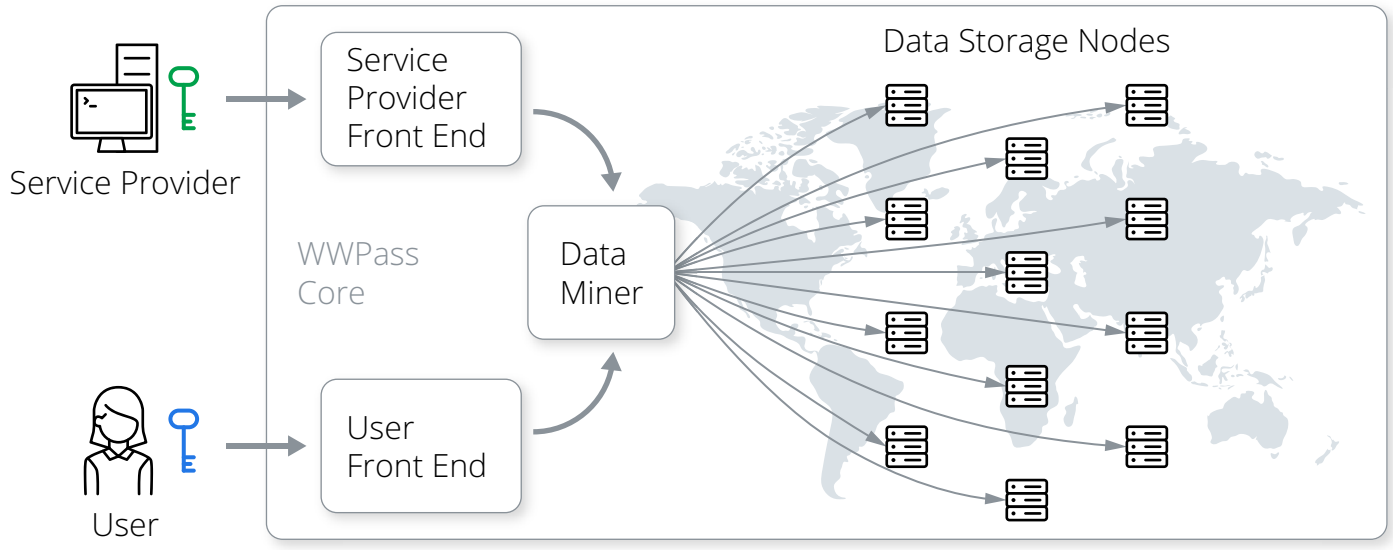


Every data container is converted into twelve pieces and stored at twelve different data centers. The redundancy implemented by a data dispersion algorithm allows WWPass to restore user data if at least any six out of twelve locations are available. At the same time, if a hacker managed to gain access to fewer than six chunks of information, it would be impossible to restore any of the original data. WWPass uses the Reed-Solomon redundancy code (6,12), which efficiently disperses information for security, load balancing and fault tolerance. See M. Rabin “Efficient dispersal of information for security, load balancing, and fault tolerance” (http://dl.acm.org/citation.cfm?id=62050).

WWPass core network structure

Digging deeper into the WWPass core network, the following components can be found:

- Service provider front ends (SPFEs) and user front ends (UserFEs) are network entities responsible for service provider and user authentication as well as user PIN verification. The SPFE transmits the data container contents between the service providers and WWPass storage.
- Data miners communicate with storage nodes, dispersing data containers on write commands and assembling them back on read commands. Data miners are responsible for preserving data integrity, error detection and correction.
- Storage nodes are a set of geographically dispersed servers that store data.
- There may be multiple front ends and data miners running concurrently, making it possible to implement a reliable system without any single point of failure.



PIN as a second authentication factor

Authentication strength is improved with additional factors. If a service provider requests it, WWPass can require users to enter a PIN in addition to their PassKey. As users provide their PIN only to WWPass, not to the service provider, they don't need to create and remember many passwords. It is similar in functionality to a bank card PIN, where the PIN refers to the card, not to the ATM. According to the zero-knowledge principle, WWPass cannot assist a user if they forget their PIN. Using both their PassKey and their Service Key, a user can reset the forgotten PIN.

WWPass employs the SRP (Secure Remote Password – http://srp.stanford.edu) protocol. The protocol is widely used for strong network authentication and is described in RFC-2945 and is ISO - standardized (ISO/IEC 11770-4).

User consent and control

According to Kim Cameron's article, The Laws of Identity (http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf), "Technical identity systems must only reveal information identifying a user with the user's consent."

WWPass follows this principle that every act of user identification needs full consent and control of the user, and the user should never doubt their information goes to the right place. The service provider name is presented to the user in every WWPass transaction and the transaction only starts on explicit user confirmation.

Authentication ticket: how to associate the service provider and user

In order to provide a proper data container, WWPass needs to associate a particular user with a specific service provider. To do this, WWPass employs a temporary transaction identifier, or ticket. The ticket combines a long random nonce (number used once) with the service provider name and some housekeeping information. Here's how it works:

1. The ticket is issued by the WWPass core service upon the request of the service provider, who then transmits the ticket to the user.
2. This ticket is downloaded into the PassKey. The user sees the service provider name, which is a part of the ticket, on the authentication dialog box and confirms the transaction.
3. The PassKey starts the authentication process with the WWPass core network. Upon success, an encrypted communication channel is created between the PassKey and the UserFE.
4. The ticket and the UserID are transmitted secretly using this secure channel back to WWPass. WWPass determines the SPID, which corresponds to a partucular transaction.
5. With both the SPID and UserID keys, WWPass calculates the data container address and its encryption key and transmits the data to the service provider. WWPass authenticates PassKeys according to the GlobalPlatform specification 2.2.1 (http://www.globalplatform.org).

Zero Knowledge Policy

Zero knowledge is one of the basic principles of the WWPass architecture and is the ultimate solution to prevent insider security breaches and leaks (backdoor). By design all users are anonymous to WWPass. The user data comes from the service provider in encrypted form with keys unknown to WWPass and returns to the service provider intact. WWPass does not store its users' data in traditional database tables. Instead users' data is kept in isolated data containers, which makes access to each chunk of data impossible without providing WWPass credentials, specific to that particular data container, and thus preventing data leaks.

Zero knowledge: how to name & store data

Only when WWPass has both the UserID and the SPID, it can store and retrieve information from the data container. It is not secure to use a concatenation of the UserID and the SPID as a container name and to store the service provider data as-is because information could be harvested by hackers, exposing all user activities at various web sites. Therefore, WWPass ciphers the container name and content and disseminates it using Reed-Solomon dispersion techniques.

To protect IDs in data container names WWPass uses a transformation for identifiers. This is in fact a "one-way injective function". This means WWPass applies a specific function to the UserID/SPID combination to create a unique container identifier. "One-way" means that it is practically impossible to 'invert' the transformation and to get back the UserID and SPID.

WWPass does not use hash functions as they do not guarantee uniqueness. What is needed is a kind of "Injective' transformation. At least two functions are known to have this property. The first is a power function modulo prime number. The second is RSA public-key encryption. Provided the private key is intentionally destroyed, this encryption becomes one-way.

Both mentioned functions work in modulo arithmetic, with relatively big numbers. Treated as bit strings, those numbers hold concatenations of the UserID and SPID. Spare bits available in the strings allow WWPass to use additional parameters – e.g. container names. This provisions multiple named data containers for each UserID/SPID combination.

Encryption is essential to secure user data. While Reed-Solomon dispersion can help keep data secret (see Shamir's "How to share a secret"), WWPass also encrypts all data containers. Every data container is ciphered with a key obtained as a hash of the SPID and UserID concatenation.

WWPass continuously checks the integrity and corrects errors without any knowledge of the customer's data. WWPass also iterates over available cryptic names of containers to determine if a chunk is lost on a particular storage node and restores it properly using the Reed- Solomon algorithm.

Zero knowledge: I lost my key (and I forgot my PIN)

User anonymity comes at a price: WWPass can't do anything when a user loses their PassKey or forgets their PIN. This leaves the user fully responsible for the safe backup of their WWPass membership, and should a PassKey be lost, all data contained by WWPass is lost forever. Therefore, WWPass provides the user with two types of keys: the first is the PassKey for everyday use, while a second Service Key is used only for key management. The user can use the Service Key to revoke or disable a lost PassKey, create a new PassKey or reset and redefine their PIN.

Applications: how to use WWPass technology

WWPass concentrates its efforts on secure authentication and storage technologies and holds several important patents for this technology. However, WWPass has also a number of out-of-the-box solutions, which include:

- Authentication modules for WordPress, Drupal, Redmine, Jenkins, Sugar CRM etc
- VPN and Remote Desktop logins
- Cloud services access - Google, Salesforce, Amazon Web Services
- encrypted cloud storage of sensitive data
- full PKI implementation for signed and encrypted email, Winlogon and disk encryption

The variety of applications is possible due to the low-level binary nature of WWPass architecture. WWPass provides a robust and easy-to-use SDK to allow third-party service providers and developers to implement WWPass support in their applications.

Ready to learn more?

When it comes to learning more about building a solution for your organization, we're here to answer your questions, large or small — just visit wwpass.com today, or contact **info@wwpass.com** or **1 (888) WWPASS1** for a personalized consultation with an expert from our integration team.

Want to dig a little deeper into the technical details? You can find a wealth of documentation about WWPass solutions, including whitepapers and datasheets, at wwpass.com/resources/documentation.

WWPass Corporation

1155 Elm Street
Manchester, NH 03101

1 (888) WWPASS1
or 1 (603) 836-4932

info@wwpass.com
wwpass.com