VASCO
THE AUTHENTICATION COMPANY

Vacman Controller

Integration Guide - White Paper

Revision 3.0

Disclaimer of Warranties and Limitations of Liabilities

The Product is provided on an 'as is' basis, without any other warranties, or conditions, express or implied, including but not limited to warranties of merchantable quality, merchantability of fitness for a particular purpose, or those arising by law, statute, usage of trade or course of dealing. The entire risk as to the results and performance of the product is assumed by you. Neither we nor our dealers or suppliers shall have any liability to you or any other person or entity for any indirect, incidental, special or consequential damages whatsoever, including but not limited to loss of revenue or profit, lost or damaged data of other commercial or economic loss, even if we have been advised of the possibility of such damages or they are foreseeable; or for claims by a third party. Our maximum aggregate liability to you, and that of our dealers and suppliers shall not exceed the amount paid by you for the Product. The limitations in this section shall apply whether or not the alleged breach or default is a breach of a fundamental condition or term, or a fundamental breach. Some states/countries do not allow the exclusion or limitation or liability for consequential or incidental damages so the above limitation may not apply to you.

Copyright

Trademarks

VASCO®, Vacman®, IDENTIKEY®, aXsGUARD™, DIGIPASS®, and ® are registered or unregistered trademarks of VASCO Data Security, Inc. and/or VASCO Data Security International GmbH. in the U.S. and other countries. All other trademarks or trade names are the property of their respective owners.

Vacman

Table of Contents

1 Overview

The Vacman® Controller allows you to integrate DIGIPASS® strong authentication inside your application replacing the insecure static passwords. By doing so, you are able to use DIGIPASS Strong Authentication and Electronic Signatures inside your application. This will help you make your critical business application more secure.

Whether you have a remote banking application (Internet Banking, Phone Banking), an authentication server that requires the usage of One Time Passwords or any other application that needs higher security, the Vacman Controller is the right answer.

This White Paper guides you through the different steps that are required to integrate the Vacman Controller inside your application. There is no need to break open your application or rewrite your application from scratch. After reading this document you will understand how the Vacman Controller interacts with your application and how it allows you to use the DIGIPASS dynamic passwords inside your application.

The Vacman Controller integrates natively inside your application, so there is no need for additional server(s), databases or hardware. Through its C- and Java-API (and .NET API on Microsoft Windows), the Vacman Controller can be integrated in any application. The Vacman Controller is available on a large variety of Operating Systems, including Microsoft Windows NT/2000/XP/Vista, SUN Solaris, AS/400 etc.

The Vacman Controller is easy to integrate, allowing your developers to focus on the application itself.

2 Problem Description

Implementing adequate security for your applications can become a difficult task that could be very time consuming. Most likely, this requires the installation of additional hardware (servers), software and changes be made to your applications.

Implementing Strong User Authentication requires modifications to different parts of your application including the password verification routines, user management and administrative tasks inside your application.

In most situations, you will end up with 2 separate databases, the application database and the authentication server database, which will bring a number of additional administrative tasks and additional costs. The user management will become more difficult, since you will need to perform user management inside your application and in the authentication server. This makes the logistics a burden once you start deploying an application on large scale.

By using the Vacman Controller, VASCO® helps you address all these problems. The Vacman Controller integrates natively inside your application. As such, you have full control over the user management, security parameters etc giving you maximum flexibility. The Vacman Controller makes your security system reliable and helps you lower the Total Cost of Ownership of your security infrastructure.

3 Concept

First we will discuss in general how user-based security is implemented inside applications. Based on this we will look closer into how the Vacman Controller integrates inside your application, to obtain a higher level of security.

3.1 User authentication traditionally

Most applications have user authentication mechanism built-in. The user authentication is typically based upon user-id and (static) password. To manage the users for your application a number of services have to be made available. Besides the actual password verification, there is the need for sophisticated user management (based upon groups/templates) and user rights have to be specified (authorization). Schematically this can be represented as follows.

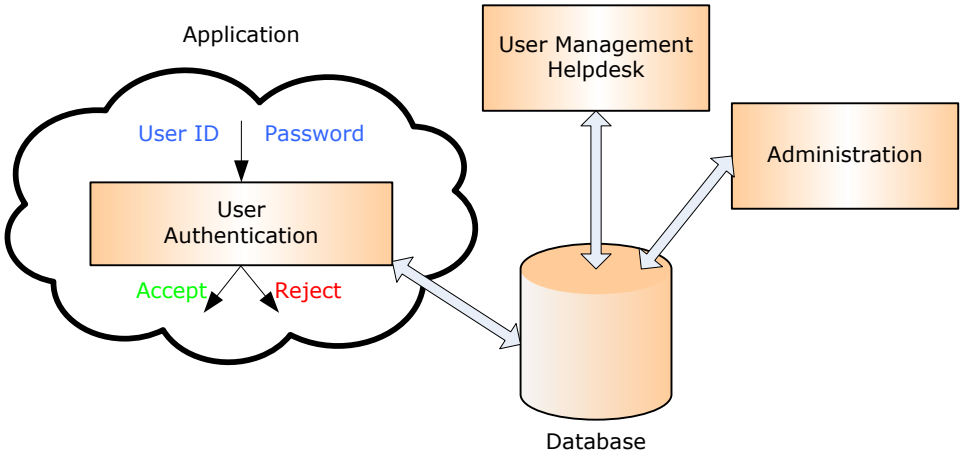


Figure 1 - Schematic overview of user level security inside application

3.1.1 User management

The user management allows you to add new users to the system, specifying user information, user-id and the password. Together with that, the users can be given specific rights for the system. The user management is often based upon groups/templates, where the user rights will be specified.

The access to the "user management" itself must be secured, to prevent unauthorized access to the system, making changes to accounts, creating additional accounts and breaking your security as such.

3.1.2 User authentication

Inside the application, a certain module is responsible for checking the user-id and the password that were given by the end-user. The verification of the user-id and the password can be based upon the comparison of the given static password with a password stored inside the database, or with a stored hash code of the password. If the password matches the information found inside the database, then the User Authentication module returns a YES/NO, indicating whether User Authentication verification was successful or not. Based upon this return code, the user will be granted access to the application or not.

3.1.3 Administration

Inside the application a number of administrative tasks can be found. These include password management rules e.g. users must change their passwords every 45 days or passwords need to be at least 6 characters long.

These administrative parameters define the general security rules of your application, thereby enforcing the required security.

3.2 Vacman Controller Integration

By integrating the Vacman Controller inside your application, one can achieve a higher level of security. This will allow you to replace the static passwords with dynamic passwords. The integration of the Vacman Controller will take place at several levels of your application. The actual integration process can be split up into the following steps.

- DIGIPASS data model integration
- DIGIPASS assignment
- Password verification
- Advanced DIGIPASS management

Only the first 3 steps are required – the last step, Advanced DIGIPASS management is optional.

3.2.1 DIGIPASS Data Model Integration

To verify the DIGIPASS dynamic password inside your application, it is required that the DIGIPASS secret keys are available at the server side. When receiving DIGIPASS, you also receive a Transport Key(often referred to as Database Key or so-called DPX Key), that contains the DIGIPASS secret keys. This database is encrypted with a 'Transport Key' that is sent to the customer. Inside the application database, one must store the DIGIPASS secret information. This DIGIPASS secret information is extracted from the DPX file, by using the DPX-import function calls that are available in the Vacman Controller.

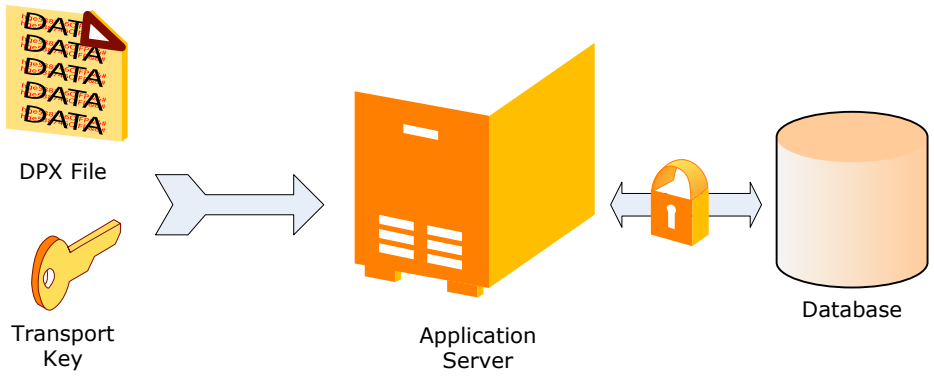


Figure 2 - DIGIPASS activation

The DPX file contains the DIGIPASS information. For each DIGIPASS application, parameters such as response length, challenge length, unlock information are stored inside the DPX file. One DPX file can contain multiple tokens, depending on how the batch (group) of DIGIPASS tokens was created.

By importing the DIGIPASS from the DPX file, you will obtain a list of different DIGIPASS inside your application database. Every DIGIPASS is referred to by its unique serial number. For every token/application that is found inside the DPX file, 4 fields are returned that must be stored inside the application database.

Importing the DIGIPASS from the DPX file is an Administrators task that must be made available from the application console or from the host machine. Whenever new DIGIPASS are programmed, the corresponding DPX file has to be uploaded into the application server database.

3.2.2 DIGIPASS assignment

Once all the DIGIPASS are imported into the Application server database, one must have the possibility to assign a DIGIPASS to a user. This task will be performed from the user management of the application.

Inside the database it is required that a link is created between the user (user-id) and the DIGIPASS. This can be done by creating an additional table that defines the link between the user and the DIGIPASS (by referring to the unique serial number of the DIGIPASS).

Depending on the design of the application, it is possible to assign only 1 DIGIPASS to 1 user, assign 1 DIGIPASS to multiple users or assign multiple DIGIPASS to 1 user.

In most situations, it is advised to assign 1 DIGIPASS to 1 user because the DIGIPASS is used to uniquely identify a certain user.

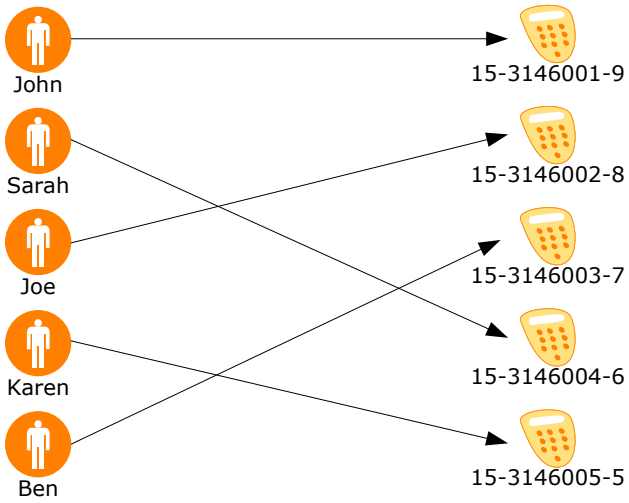


Figure 3 - Assign DIGIPASS to users in application database

3.2.3 Password verification

To complete the integration, it is required that changes are made to the module where the password verification is performed. At this point in the application the following steps are required:

1. Check the database to find the DIGIPASS that is assigned to the user and retrieve the corresponding DIGIPASS data.

2. Check in the DIGIPASS data whether Challenge/Response is used. If yes, generate the Challenge, update the DIGIPASS data and send the Challenge to the user.

3. Call Vacman Controller function to verify the users dynamic password

4. Update DIGIPASS data

5. Accept or Reject user based upon response from the Vacman Controller

Remarks:

- A function call is made available from the Vacman Controller that allows you to generate a random Challenge code, used in the Challenge/Response algorithm. If required, you can decide to use your own Challenge generation mechanism.

- To verify a DIGIPASS dynamic password, there exists only 1 function call inside the Vacman Controller. This very same function is used for all possible DIGIPASS algorithms, whether you are using Time-Based, Event-Based, Challenge/Response or a combination of the previous.

- The Vacman Controller comes as static or shared library that will act as a part of the application. There is no need for additional hardware or software to be installed.

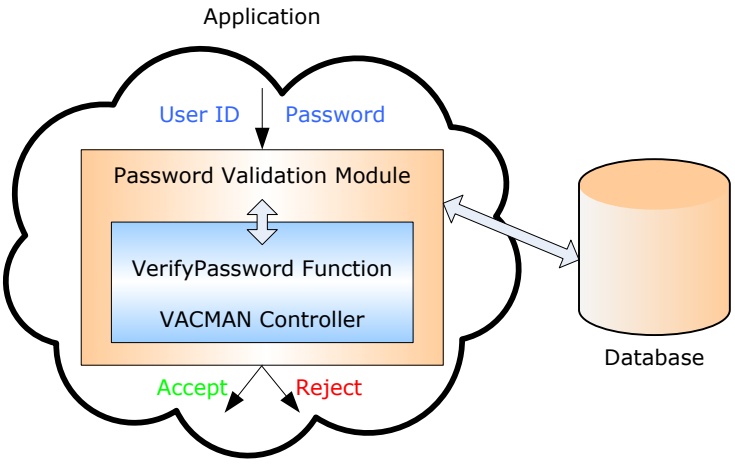


Figure 4 - Vacman Controller interaction with the password verification module inside application

3.2.4 Advanced DIGIPASS management

The 3 steps mentioned before are required for an integration of DIGIPASS One Time Password mechanism inside an application. Optionally, some more advanced tasks can be implemented.

The DIGIPASS allows remote unlocking; in the case an end-user has locked the DIGIPASS after entering a number of wrong PIN codes. To unlock a DIGIPASS, its secret unlock key has to be known on the server side. By using the Vacman Controller import routines, the unlock key is transported from the DPX file to the application database. The Vacman Controller has a function call that allows you to generate the unlock response for a certain DIGIPASS given the unlock challenge. This type of function can be made available to your helpdesk operators as part of the helpdesk application.

The Vacman Controller also holds a token reset function that will allow you to reset a number of token specific parameters.

- Identification/Signature error counter
- DIGIPASS time drift

The Vacman Controller has an Identification/Signature threshold that will be triggered after a number of successive wrong Identification or Signature codes. By calling the Reset function, these error counters are reset.

The Vacman Controller has automatic clock drift tracking functionality for every individual DIGIPASS. This allows you to track the time drift of the DIGIPASS internal clock (in case you are using Time-based algorithms). By calling the Reset function, one can reset the previously measured token time drift, allowing for automatic time drift synchronization at the next password verification.

3.2.5 DIGIPASS Digital Signature

The Vacman Controller offers support for DIGIPASS Digital Signature.

The DIGIPASS has the ability to generate a Digital Signature on a number of data fields, e.g. transaction data. This is particularly interesting for Remote Banking applications. Apart from consulting data inside the banking application, more and more banks are offering the ability to transfer funds or money.

Whenever someone performs a transaction, a certain number of data fields are critical in the transaction. Typically these are debit account number, credit account number and the amount of money.

When the end-user is performing a transaction, he will be requested to enter the data inside the DIGIPASS, which will generate an Electronic (Digital) Signature or MAC code (Message Authentication Code) based upon the data, (for detailed information please look at the VASCO DIGIPASS Family of Tokens Technical White Paper). The Digital Signature will be sent along with the transaction data and will be verified at the server side. In the event someone tampered with the transaction data, the previously generated Digital Signature will no longer match. Based upon this information, the transaction can be rejected at the server side.

The Digital Signature verification function is a standard function that is available in the Vacman Controller.

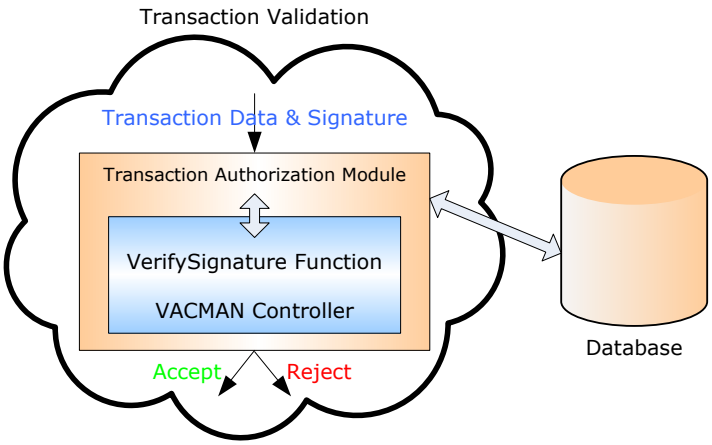


Figure 5 - Vacman Controller interaction with the Transaction verification module inside TRANSACTION Server

3.2.6 Virtual DIGIPASS integration

The Virtual DIGIPASS intends to (temporarily) replace the traditional DIGIPASS. The user who wants to use the Virtual DIGIPASS, requests the delivery of a One Time Password through a channel (e.g. Web-browser), where, after successful validation of the users credentials (e.g. username/static PIN-password or other information), the host-system takes the initiative to generate a Virtual One Time Password, which is then provided to the customer via another channel, typically SMS, Text Message or other.

The VASCO Virtual DIGIPASS is available as a Backup solution or as a Primary solution.

The Backup Virtual DIGIPASS (BVDP) is typically used in situations where a regular user has lost or forgotten his traditional DIGIPASS. Through a simple operation on the helpdesk (if required), the Backup Virtual DIGIPASS is enabled for this individual user, this to limit the usage of the Virtual DIGIPASS in time or credits.

The Primary Virtual DIGIPASS (PVDP) is typically used in situations where a user occasionally needs to logon to the application (e.g. a few times a year).

The Primary and Backup Virtual DIGIPASS are typically positioned for occasional use, mainly because of economical reasons – every authentication requires an SMS or Text message to be delivered.

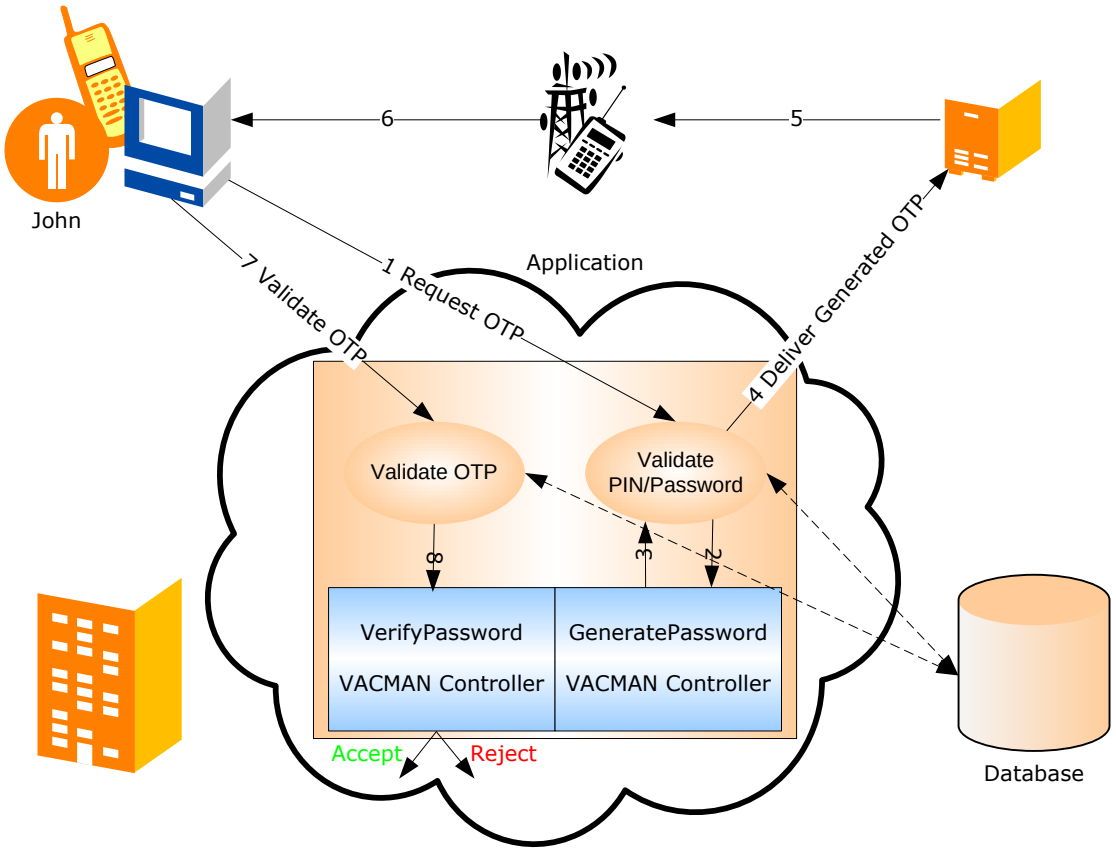


Figure 6 - Virtual DIGIPASS

The authentication flow is as follows:

1. User requests a Virtual One Time Password (e.g. through web-browser)
2. After validating the users' credentials against the database, the Vacman Controller is requested to Generate a Virtual One Time Password
3. The Vacman Controller returns the Virtual One Time Password to the application
4. The application provides the Virtual One Time Password to an SMS Gateway
5. The SMS Gateway provides the SMS containing the Virtual One Time Password to Mobile network
6. The SMS containing the Virtual One Time Password is transmitted to the users mobile.
7. The users types his username and the received Virtual One Time Password in the password field of the application
8. The application passes the Virtual One Time Password to the Vacman Controller which verifies this

The generation of the One Time Password is only allowed if this function has been enabled for this individual DIGIPASS.

The validation of the Virtual One Time Password is done with the exact same function calls as with the traditional DIGIPASS One Time Password validation – hence the application will not notice any difference between a traditional or Virtual DIGIPASS One Time Password.

3.3 Key advantages

3.3.1 Easy to integrate

The required steps to integrate the Vacman Controller inside your application are reduced to a minimum. As a result, this allows you and your developers to focus on the business part of the application, not having to worry about the details how dynamic passwords are managed and used inside your application. The Vacman Controller integrates seamlessly inside your application, thereby giving you full control over user management, DIGIPASS management and the applied security rules.

3.3.2 Platform support

The Vacman Controller is developed to have support for different platforms. Through the C, Java and .NET API the Vacman Controller integrates with any application. The Vacman Controller uses the same paradigm across all supported platforms, allowing you to scale your security infrastructure with your business requirements. You can begin deploying your application on low-entry systems (e.g. Linux, Microsoft Windows based) and let it grow with your business to mid-range or high-end systems (e.g. SUN Solaris SPARC, HP/UX, AS/400 based). Since the Vacman Controller API is platform independent, the very same integration method is used across all these platforms.

3.3.3 High security

Integrating the Vacman Controller inside your application allows you to enforce a high level of security by removing the security flaw related to static passwords (PIN). The static passwords will be replaced with dynamic passwords (Time-based, Challenge/Response) giving you higher security and relieving you from the burden of managing the users static passwords. Additionally you can use the DIGIPASS Digital Signature functionality inside your application.

The Vacman Controller provides you with strong encryption and integrity control on the DIGIPASS information that is stored inside the application database. The Vacman Controller uses Triple DES encryption algorithm for the data, based upon keys that are managed by the integrator or customer. As such VASCO guarantees a complete security chain from the moment the DIGIPASS is programmed until it is activated and used in the application.

3.3.4 DIGIPASS Family enabled

The Vacman Controller is the preferred way to integrate the DIGIPASS technology inside your application. This allows you to use any of the DIGIPASS Family members for your security. You are free to combine the use of any of the DIGIPASS Family members, making sure that you can choose the best solution for your customers.

For more information on the DIGIPASS Family, please refer to the ‘VASCO DIGIPASS Family of Tokens Technical White Paper’.

4 Technical Description

The following paragraph gives you detailed information on how the Vacman Controller integrates inside your application. We will discuss this integration by working out the different steps that are required for the integration.

4.1 DIGIPASS Data model integration

4.1.1 DIGIPASS data structure

To use the Vacman Controller a number of data fields must be stored inside the application database. These fields are required for the dynamic password verification. For every DIGIPASS application the following 4 fields are used.

- Serial number + Application name – 22 characters
- DIGIPASS type – 5 characters
- Mode – 2 characters
- DIGIPASS data – 248 base64 encoded data

Serial number + Application name

This field contains the unique DIGIPASS serial number, concatenated with the name of the application how it is referred to inside the DPX file. The DIGIPASS can be programmed to hold multiple applications (e.g. Logon, Digital Signature). Each of those applications can hold a unique set of parameters and keys. Each of those applications is stored separately.

DIGIPASS type

This field describes the type of DIGIPASS. Possible values are DP300 (DIGIPASS 300), DPGO3 (DIGIPASS GO3)

Mode

This 2-character field describes the mode of operation of this DIGIPASS application. The following values are accepted:

- RO: Response Only
- CR: Challenge/Response
- SG: Electronic Signature
- MM: Multi-Mode application

DIGIPASS data

This field contains all the DIGIPASS application specific parameters, such as the Triple DES keys that are used, unlock key, response information and challenge information (when applicable). This data block is encrypted using Triple DES encryption algorithm.

Preferably, the 4 above-mentioned fields are stored inside a separate DIGIPASS table inside the application database.

4.1.2 DIGIPASS data import

The Vacman Controller provides you with import functionality for the DIGIPASS, based upon the DPX file that is corresponding to the programmed DIGIPASS.

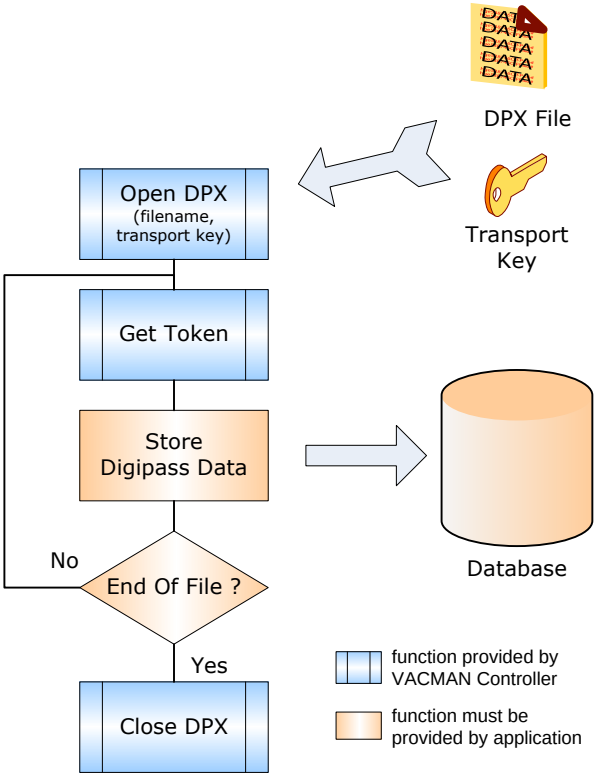


Figure 7 - Import DIGIPASS data flowchart

The DIGIPASS import procedure is as follows:

1. Open the DPX file by calling the 'Open DPX' function from the Vacman Controller. This function requires the DPX file name and the database key as input parameters. The operator will enter these parameters into the system when uploading new DIGIPASS.

2. Call the 'Get Token' function from the Vacman Controller. This function returns you the 4 fields specified in the previous paragraph.

3. Store the 4 fields inside the DIGIPASS table of the application database

4. Check whether the retrieved token is the last token inside the DPX file

5. If YES, call the 'Close DPX' function from the Vacman Controller

6. If NO, go back to step 2

The DIGIPASS import functionality is an administrative task in the application. For every batch of DIGIPASS, one will receive a DPX file, with the corresponding database encryption/decryption key. Every DPX file needs to be uploaded only once into the system.

4.2 DIGIPASS assignment

In the previous step we discussed how the DIGIPASS information is uploaded to the application database. As such, you will obtain a list of DIGIPASS, based upon the serial number of the DIGIPASS that you can assign to the different users.

Assigning a DIGIPASS to a user is a task that can be performed from the user management of the application. Technically speaking, it is sufficient to create a link between the user table and the DIGIPASS table inside the application database.

Different relationships are possible between the user table and the DIGIPASS table. This can be entirely chosen when designing the application data model. Most common, a one to one relationship is used for assigning the DIGIPASS to a user. As such, one has the ability to identify every user individually.

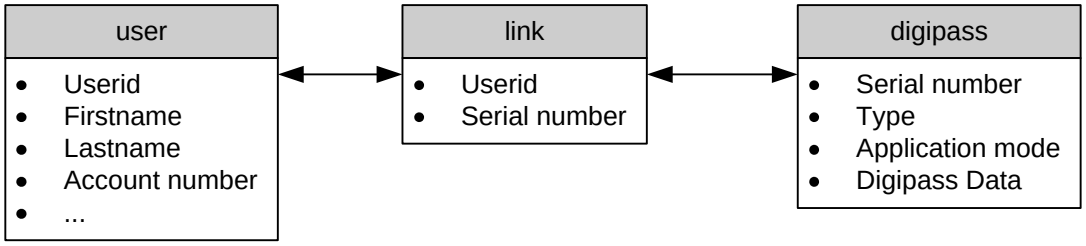


Figure 8 - Relationship between Users and DIGIPASS

4.3 Password verification

After integrating the basic administrative tasks such as DIGIPASS import and DIGIPASS assignment, it is necessary to integrate the password verification routines. The Vacman Controller needs to be integrated in the module where the user-id and static passwords are verified.

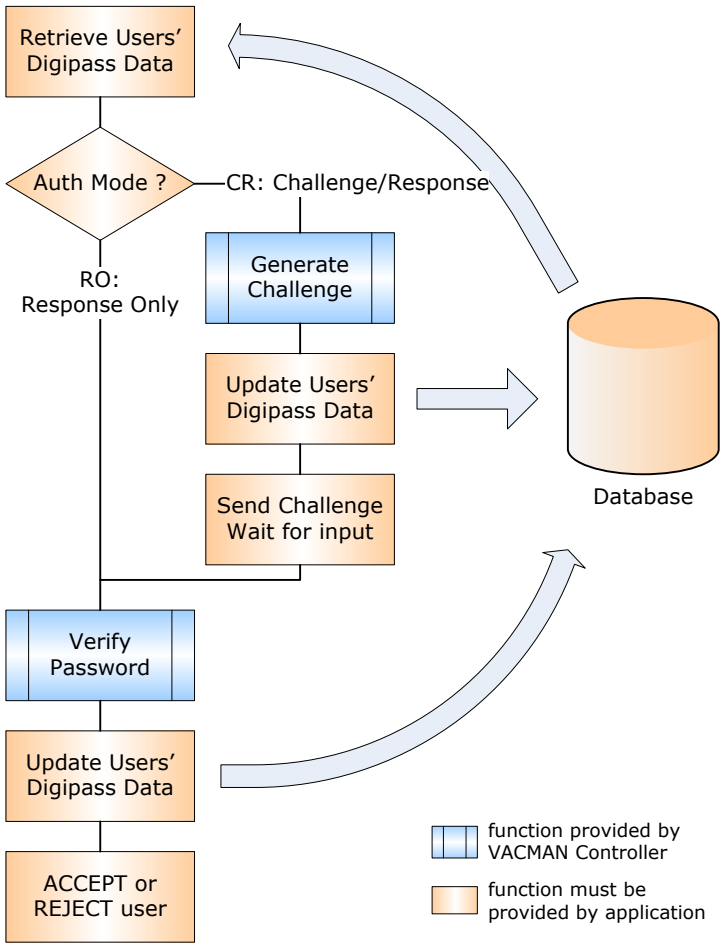


Figure 9 - Password verification routine

The password verification procedure is:

1. Check the link database to find the serial number of the token assigned to the user and retrieve the corresponding DIGIPASS data

2. Check the Authentication mode used. When using Response Only (RO) go directly to 3.

    a. When using Challenge Response (CR), call the 'Generate Challenge' function from the Vacman Controller

    b. Update the DIGIPASS data to allocate the generated challenge to this DIGIPASS

    c. Send Challenge to user and wait for response on the challenge

3. With the users response (including the challenge if required) call the 'Verify Password' function from the Vacman Controller

4. Update the DIGIPASS data in the application to store date/time DIGIPASS was last used, time synchronization…

5. Based upon the return code of the Vacman Controller Verify Password function, the user will be accepted or rejected

4.4 Digital Signature verification

Integrating the Vacman Controller inside your application allows you to use the DIGIPASS Digital Signature capabilities. The Digital Signature creates a higher level of security inside your application, by proving the identity of the user that has generated the signature and protecting the content of the transaction. The Vacman Controller has the functionality to verify such Digital Signature inside the application.

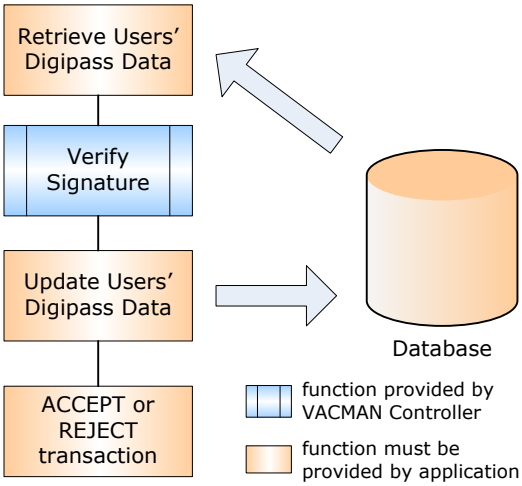


Figure 10 - Digital Signature verification routine

The Digital Signature verification procedure is:

1. Check the link database to find the serial number of the token assigned to the user and retrieve the corresponding DIGIPASS data

2. With the transaction data and the Digital Signature given by the user, call the 'Verify Signature' function from the Vacman Controller

3. Update the DIGIPASS data

4. Based upon the return code of the Verify Signature function, the transaction will be accepted or rejected

4.5 DIGIPASS Unlocking

After entering a number of invalid PIN codes, a users' DIGIPASS can lock. The DIGIPASS has the unique capability to be unlocked remotely. When the DIGIPASS locks, the unlock challenge will be shown on its display. Based upon this unlock challenge, an unlock response must be generated. This unlock response can be generated from the Vacman Controller, based upon the information stored inside the application database. For detailed information about the DIGIPASS Unlocking capabilities, please refer to the DIGIPASS Deployment Procedures Technical White Paper.

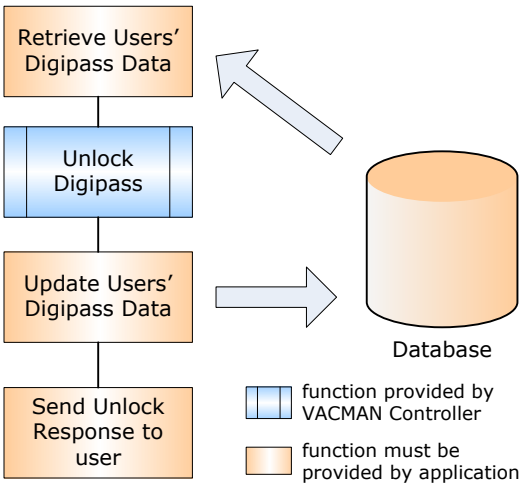


Figure 11 - Generate DIGIPASS unlock response

The DIGIPASS Unlock code generation procedure is:

1. Check the link database to find the serial number of the token assigned to the user and retrieve the corresponding DIGIPASS data

2. Call the 'Unlock DIGIPASS' function from the Vacman Controller, given the DIGIPASS Data and the unlock challenge

3. Update the DIGIPASS data

4. Return the generated unlock response to the end-user

4.6 DIGIPASS Reset

The Vacman Controller holds a number of internal parameters for every DIGIPASS.

- Error counter for number of successive invalid Identification/Signature codes
- Token time deviation
- Number of inactive days since the last valid Identification or Signature

This information can be reset by calling the 'Reset Token' function from the Vacman Controller.

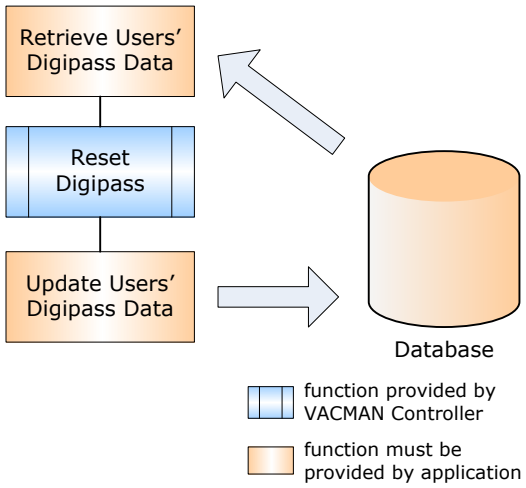


Figure 12 - Reset DIGIPASS information

The Reset DIGIPASS Information procedure is:

1. Check the link database to find the serial number of the token assigned to the user and retrieve the corresponding DIGIPASS data
2. Call the 'Reset DIGIPASS' function from the Vacman Controller, given the DIGIPASS Data
3. Update the DIGIPASS data

Note:

The Reset can be used for 3 reasons:

- The user has reached the maximum threshold counter after successive erroneous attempts.

- The DIGIPASS clock has drifted too far since the last successful authentication. The DIGIPASS is out of sync.

- The number of allowed inactive days has been reached.

4.7 Virtual DIGIPASS integration

The Vacman Controller has a feature allowing you to generate a Virtual DIGIPASS One Time Password (given the function was enabled for this DIGIPASS). Typically, the Virtual DIGIPASS One Time Password will be generated, after the user has been identified in the application.

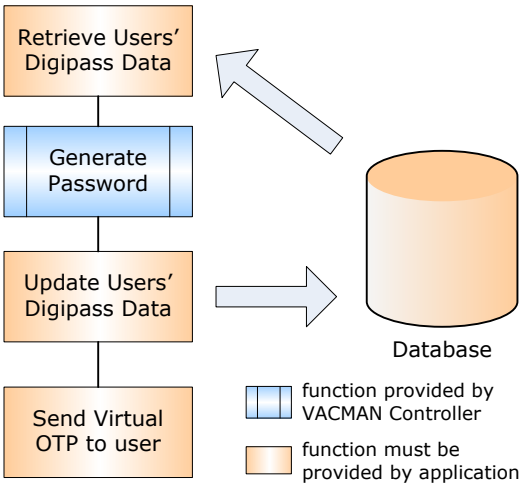


Figure 13 - Virtual DIGIPASS Integration

The Vacman Controller Generate Password function returns you the generated Virtual DIGIPASS One Time Password, which can be provided to the user, e.g. via SMS or Text based service.

Upon the next password validation, the Vacman Controller will then accept the Virtual DIGIPASS One Time Password as it would have been generated by a hardware DIGIPASS.

5 Vacman Controller Features

5.1 Secure chain from programming to verification

VASCO ensures an entirely secured chain, starting from the DIGIPASS Programming, either performed by VASCO in the factory or Logistics department. Alternatively the DIGIPASS Programming can be performed by the customer itself or a 3rd party services/provisioning company.

5.1.1 Transporting the DIGIPASS information

To guarantee the secure transport of the DIGIPASS Keys, VASCO uses the highly secured DPX file. The DPX file contains all the DIGIPASS specific data – profile (codeword), application information, keys - allowing the verification of the DIGIPASS OTP on the host system. The security sensitive information in this DPX file is encrypted using a 3DES key, which is derived from a 3DES transport key. The transport key is shipped via separate way to the security officer, using a sealed PIN letter. A wide variety of alternatives is available:

- Splitting the transport key in smaller parts, shipping them to different persons, via different channels (e.g. E-mail, Fax, PIN letter)
- Assembling the transport key from a number of transport key components – each of these is a full 3DES key
- Applying additional encryption to the DPX file (e.g. PGP)

More details on the different transport methods for the DIGIPASS information is available upon request.

5.1.2 DIGIPASS Data Encryption

Through the DPX file, the DIGIPASS information is transported in a secure way from the programming station to the host system where the validation takes place. On the host system, the confidentiality of the DIGIPASS data must be ensured as well. For this purpose, the Vacman Controller uses a 3DES storage key to encrypt the DIGIPASS data with. The storage key is different for each DIGIPASS data record. The storage key is generated based upon a 3DES key inside the Vacman Controller (a key controlled by VASCO), a 3DES key which can be provided by the customer through the API and the serial number of the DIGIPASS. Only by combining these different elements one will be able to successfully access the DIGIPASS information.

Additionally, the DIGIPASS data record also contains a checksum value, which ensures the integrity of the DIGIPASS data. The checksum incorporates all critical DIGIPASS data, such as keys, serial number, error counter etc. Anybody tampering with the DIGIPASS data block directly typically will not benefit from this since all subsequent verification calls will fail with a checksum error.

More detailed information is available upon request.

5.1.3 Optional HSM Integration

The Vacman Controller optionally can be integrated with a Hardware Security Module (HSM), to ensure that the DIGIPASS keys are never exposed in the clear.

While the standard DIGIPASS data encryption typically provides customers already with a sufficient level of security, VASCO recognized the need for higher level of security, typically in large-scaled deployments and high-security environments.

Vacman Controller can integrate with a number of industry leading HSM modules, including Thales, nCipher, Safenet and IBM ICSF. For more information about the Vacman Controller HSM Integration, please refer to the Vacman Controller HSM Integration White Paper.

5.1.4 Optional CTVS Support

The Vacman Controller can be used optionally with EMV CAP applications.

The Vacman Controller for CTVS is an authentication library offering the EMV CAP Token Validation Service combined with standard Vacman Controller authentication services (DIGIPASS, Virtual DIGIPASS and OATH).

For more information about the Vacman Controller for CTVS, please refer to the Vacman Controller for CTVS Integration White Paper.

5.2 Digipass Time drift management

Vacman Controller has a built in mechanism that will automatically manage the clock drift for every individual DIGIPASS.

The usage of time-based algorithms has a number of major benefits – the generated One Time Password can not only be used once – additionally they have a very limited life-span – thereby requiring the fraudster to immediately consume a captured One Time Password, thereby discouraging phishing attacks.

The DIGIPASS real time clock is quite accurate, however time-drift of the DIGIPASS can not be predicted, since this is influenced by environmental conditions such as temperature, humidity, batter level etc. Therefore a robust and proven mechanism is required to manage the possible time-drift of a DIGIPASS transparently on the host system.

For this purpose, Vacman Controller has the built-in concept of time windows. Vacman Controller works with 2 different time windows: a synchronization time window and an Identification/Signature time window. The time windows indicate the number of valid one time passwords at a given time. The size of each of these time windows can be configured through the Vacman Controller Kernel Parameters.

The synchronization time window is used in the following situations:

- First DIGIPASS OTP validation – to ensure both clocks are aligned
- After a call is made to the Reset Token Info function (e.g. when OTP validation fails a Reset Token Info may be performed)
- When calling the function Sync Token and Host – where the user is prompted to enter 2 consecutive One Time Password.

When the DIGIPASS OTP verification takes place and the synchronization window is used, the synchronization window will be positioned centrally at the current time and the DIGIPASS time deviation information will be updated to reflect the measured time drift.

The Identification/Signature time window is used for the normal/operational OTP verification. When the DIGIPASS OTP verification takes place and the Identification/Signature window is used, the Identification/Signature time window is positioned centrally at the current time corrected with the previously measured time deviation and the DIGIPASS time deviation information will be updated to reflect the measured time drift. Vacman Controller has a built in advanced mechanism to avoid the introduction of fake/false time-drifts (e.g. when a user explicitly waits longer to enter the OTP).

The size of the Synchronization and Identification/Signature time windows are determined by the customer, integrating the Vacman Controller – typical values are provided for each different algorithm that is supported by the DIGIPASS. For more information please refer to the DIGIPASS Algorithms document.

5.3 Dynamic time window

In high security environments, VASCO has recognized the need for using Time-based algorithms with a very small granularity (frequency of changing the OTP). The DIGIPASS support time-based algorithms with a granularity up to 8 seconds. Together with this small granularity, there is also the need to work with extremely small windows of acceptance.

To find the optimum between security (using a small window of acceptance), user friendliness (avoiding false rejects caused by time deviation) and cost (false rejects will lead to helpdesk calls), the Vacman Controller has built-in the concept of Dynamic Time Windows, which can be enabled by the customer if desired. The dynamic time-window allows you to work with extremely narrow time-windows; however, it will automatically enlarge the Identification/Signature time window, depending on the number of days since the previous OTP validation happened. More detailed information is available upon request.

5.4 Authorize unlock

The Vacman Controller has a built-in authorization mechanism, allowing the split-up of unlock responsibilities at the helpdesk. In case a user has a locked DIGIPASS, the user can contact the helpdesk, requesting for an “Unlock Authorization Code”, which is generated by the Vacman Controller. The user will connect to a self-help-center web-page, where he/she is requested to enter the “Unlock Authorization Code”, the DIGIPASS Serial Number and the DIGIPASS Unlock Challenge. The Vacman Controller will then verify whether the Unlock Authorization Code matches the previously generated code and then generates the unlock response for the locked DIGIPASS. With this unlock response, the user will be able to successfully unlock the DIGIPASS and choose and confirm a new PIN code to open his/her DIGIPASS. The “Authorize unlock” provides you with a higher level of security, since the operator is only able to generate the unlock authorization code and NOT the actual unlock response for the DIGIPASS.

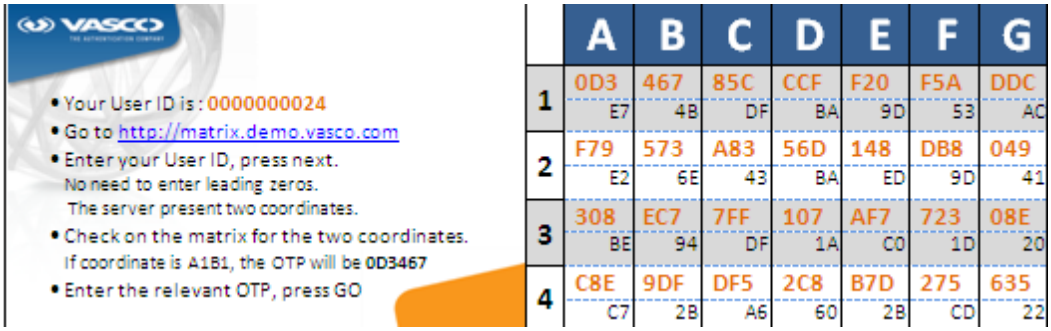
5.5 Offline signature validation

Vacman Controller has the capability to verify transaction signatures, generated by the DIGIPASS.

In case of dispute on certain transactions, Vacman Controller allows you to verify again whether a given signature code matches a transaction, signed by a specific DIGIPASS on a specific date (historical). The offline verification (or deferred signature validation) of DIGIPASS signature is also commonly used in Fax Banking applications.

5.6 DIGIPASS Matrix Cards Support

The DIGIPASS matrix card management feature is dedicated to matrix card generation and matrix card user authentication. A DIGIPASS matrix card is a card on which a matrix of OTP has been printed.



	A	B	C	D	E	F	G
1	0D3	467	85C	CCF	F20	F5A	DDC
	E7	4B	DF	BA	9D	53	AC
2	F79	573	A83	56D	148	DB8	049
	E2	6E	43	BA	ED	9D	41
3	308	EC7	7FF	107	AF7	723	08E
	BE	94	DF	1A	C0	1D	20
4	C8E	9DF	DF5	2C8	B7D	275	635
	C7	2B	A6	60	2B	CD	22

Figure 14 - Sample of matrix card

It is characterized by a number of parameters:

- The number of columns of the matrix card.
- The number of rows of the matrix card.
- The number of characters per cell of the matrix card.
- The format of the characters of a cell
- A DIGIPASS serial number, unique for every card.
- A sequence number representing a life cycle event

As the OTP are not dynamically renewed on the card, the card must be changed regularly. Each new card keeps the same DIGIPASS serial number and has an incremented sequence number.

The DIGIPASS matrix card management feature enables the generation of the OTP and (optionally) the host code to print on the card. The service also enables validation of the matrix card OTP using a challenge response authentication mode.

- To authenticate a DIGIPASS matrix card owner, the Vacman controller generates a certain number of matrix cell coordinates. This number is the security level. Then the user submits the OTP printed in the corresponding cells and the Vacman Controller verifies them. Optionally, it returns a host code, if it is printed in the card cells.

5.7 Software DIGIPASS Support

The following functionalities are dedicated to the Software DIGIPASS support

5.7.1 Software DIGIPASS Activation feature

The DIGIPASS activation feature is designed to facilitate the activation of DIGIPASS software in offline or online mode. Activating DIGIPASS software consists of putting settings and secrets in the DIGIPASS.

The service offers two modes to generate data for software DIGIPASS:

- Offline mode: in this mode, The Software DIGIPASS comes with application parameters properly configured. An activation code is generated, sent to the end user (via regular mail, email …) and entered manually into the DIGIPASS. This activation code contains the DIGIPASS secrets.
- Online mode: in this mode the full activation data are generated in batch, the activation data are generated and pushed into the DIGIPASS without user involvement. This activation data contains both application parameters and the DIGIPASS secrets.

5.8 MITMA Countermeasure

The MITMA Countermeasure is designed to prevent a Man in the Middle Attack (MITMA) with Software DIGIPASS using additional information from the network connection to authenticate the session.

A real Time Man in the Middle Attacks consists of having a machine between client and server. The user working on the client machine (after receiving a phishing email for example) is redirected to the Man in the Middle machine, believing he is on the regular banking site. The Man in the Middle (MITM) then initiates a connection to the banking site as well. From now on, all data that is exchanged between the client and the server machine is compromised. Besides violating the confidentiality of the data exchanged between the client and the server (e.g. Account statements) the Man In The Middle now also has the capabilities to modify transaction data, such as the transaction amount and receivers account, leading to fraud.

The VASCO solution is to embed the SSL public key in the DIGIPASS One Time Password (OTP) calculation. By doing this, the server can use its own public key to validate the DIGIPASS OTP. The DIGIPASS OTP can only be validated by the server on which the SSL session was opened when generating the DIGIPASS OTP. In the case there is a Man In The Middle Attack as shown in the picture below, the DIGIPASS OTP validation will fail, as the SSL public key used in the DIGIPASS OTP generation will be different from the one used for the DIGIPASS OTP validation.

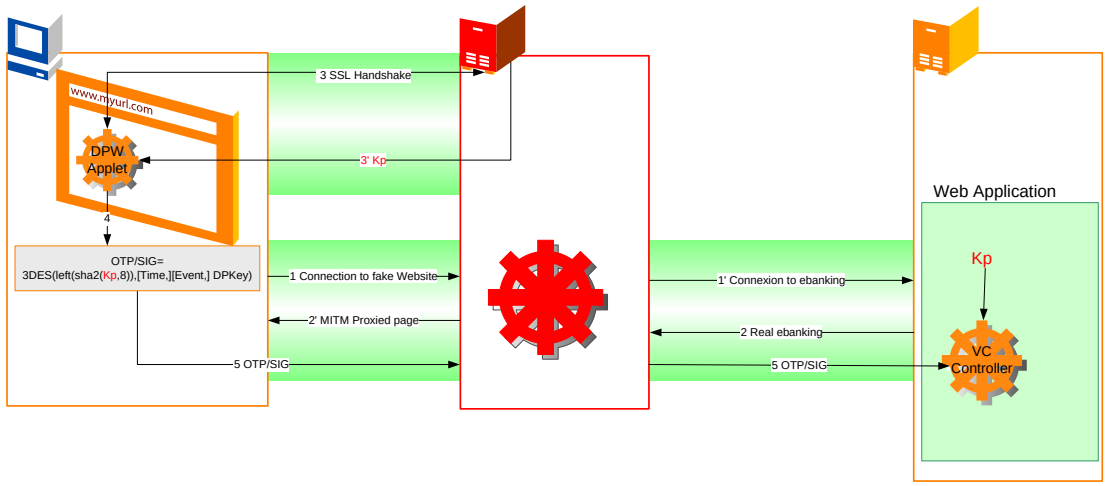


Figure 15 - MITMA Countermeasure workflow

6 Environment

6.1 Supported OS

Vacman Controller can be used on practically any platform because of the platform independence of this product. The Vacman Controller has been successfully tested and run on most common 32 and 64-bits platform.

Supported platforms are:

- Windows NT/9x/Me/2000/XP/2003/Vista
- Linux
- Sun Solaris Sparc / Intel
- HP/UX
- AIX
- AS/400
- OS/390
- Z/OS

For the latest update please contact VASCO.

6.2 Support languages

The Vacman Controller supports a wide variety of calling languages. On Windows platforms these are:

- C / C++
- Java
- C# (.NET)

On most common Unix platforms the following languages are supported:

- C / C++
- Java

On mainframe the Vacman Controller supports:

- C / C++
- Java
- COBOL
- PL1
- Assembler

6.3 Available Samples

6.3.1 Vacman Controller Integration Samples

The Vacman Controller Integration Samples (VCIS) are supplied independently from the Vacman Controller. The VCIS contain a number of sample applications for Windows platform, including C, Java, and C#.

The Java sample application can run on any platform supporting Java.

The C sample application can easily be ported to any Unix platform, by simply replacing the Windows specific database calls with the Unix variants.

On the mainframe the Vacman Controller comes with Cobol and PL1 samples.

The Vacman Controller Integration Samples are the perfect quick start to implement Vacman Controller into your application.

6.3.2 Vacman Controller Web Samples

The Vacman Controller Web Samples (VCWS) are supplied independently from the Vacman Controller. The VCWS contain samples application to demonstrate how to integrate the Vacman Controller in a web based environment. VASCO provides 2 different web applications for 2 different platforms:

- JSP/Servlet Java
- ASP/ Handler .NET

Both versions are documented for an immediate integration on a Microsoft Windows Operating System. The Java version can easily be ported to a Unix system .

The Vacman Controller Web Samples provide you with a perfect quick start to implement Vacman Controller into your own web application.

6.4 Available tools

6.4.1 DIGIPASS for C API

The “DIGIPASS for C API” is a C library that enables VASCO DIGIPASS engine in your application.

The DIGIPASS for C API offers a DIGIPASS simulation API which is easy to interface (Windows, Linux and Solaris platforms), with a single activation code (offline activation) or an Encrypted Full Activation Data (XFAD)(online activation). The DIGIPASS for C API will allow you to generate, through the API, a One Time Password or a Digital Signature, which can be used on a server application.

6.4.2 DIGIPASS for Java API

The “DIGIPASS for Java API” is a Java library that enables VASCO DIGIPASS engine in your application.

The DIGIPASS for Java API offers a DIGIPASS simulation API which is easy to interface (all platforms supporting Java), with a single activation code (offline activation) or an Encrypted Full Activation Data (XFAD)(online activation). As the DIGIPASS for C API, the DIGIPASS for Java API will allow you to generate, through the API, a One Time Password or a Digital Signature, which can be used on a server application.

6.4.3 DIGIPASS Simulator

The DIGIPASS simulator offers a DIGIPASS simulation API which is easy to interface (Windows platforms only), which comes by default with a DPX file for 100, 1000 or 10000 different DIGIPASS tokens - each with their own serial number and secrets. The API provided allows you to quickly generate different One Time Passwords for different DIGIPASS serial numbers. On the test back-end all these user accounts, linked with their respective DIGIPASS serial numbers must be created. Through the API this can be integrated into an automated test-system. As such the bank can easily test different amounts of simultaneous users logging on to the system. This is typically used for fully automated load/stress testing.

7 About VASCO Data Security

VASCO designs, develops, markets and supports patented user authentication products for the financial world, remote access, e-business and e-commerce. VASCO's user authentication software is delivered via its DIGIPASS hardware and software security products. With over 20 million DIGIPASS products sold and delivered, VASCO has established itself as a world-leader for strong User Authentication with over 440 international financial institutions and approximately 2,300 blue-chip corporations and governments located in more than 100 countries.