



VACMAN
controller

Vacman Controller

HSM Integration Guide - White Paper

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

Copyright © 2009 VASCO Data Security, Inc, VASCO Data Security International GmbH. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of VASCO® Data Security Inc.

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	Introduction.....	4
1.1	Background.....	4
1.2	Supported HSM Models	4
1.3	Glossary	4
1.4	Target Audience.....	5
2	HSM Integration Architecture.....	6
2.1	General	6
2.2	HSM Independence.....	6
2.3	Operating Modes	6
2.3.1	Data Encryption Only.....	7
2.3.2	Data Encryption and DES Operations	8
2.3.3	Data Encryption, DES Operations and VC Operations	9
3	Vacman Controller for HSM – Additional Features.....	11
3.1	DPX Double Encryption	11
3.2	HSM Storage Key Migration.....	12
3.3	HSM Load Balancing and Fail Over	12
3.4	HSM Key Names and Identifiers.....	12
4	Additional information	13
5	About VASCO Data Security.....	14

1 Introduction

1.1 Background

The Vacman® Controller allows the verification of passwords and signatures generated by a DIGIPASS®. To verify these passwords, the Vacman Controller needs access to the parameters and secrets that were programmed in the DIGIPASS. The Vacman Controller will store this information in what is called the DIGIPASS Blob. The DIGIPASS Blob is a flat data structure that must be stored in the Application Database on the Host computer that runs the Vacman Controller.

To prevent fraud from people having access to the database, the DIGIPASS Blob is protected (privacy and integrity) with a 3DES-based encryption.

The standard Vacman Controller uses a Software Security Module (SSM). This means that the DIGIPASS Keys used in the Vacman Controller verification process are encrypted by (1) secret codes stored in the software; and (2) secret codes passed during run-time. In the standard implementation of Vacman Controller, these DIGIPASS Keys are therefore, at some point during the verification process, exposed on the Application Host.

Benefits of HSM implementation

In case a higher level of security is required, or to benefit from an existing architecture, it is possible to use the "Vacman Controller for HSM" solution.

A Hardware Security Module (HSM) is a tamper-proof hardware device that is connected to, or inserted into the Application Host.

A HSM facilitates secure long-term storage for cryptographic keys, in combination with cryptographic processing capabilities. Typically a HSM Storage Key is used to encrypt the DIGIPASS Keys that are used to authenticate users and transactions. The encrypted DIGIPASS Keys remain in the DIGIPASS Blob and stored in the Application Database.

Note:

The DIGIPASS Blobs and the DIGIPASS Keys are NOT stored in the HSM – only the HSM Storage Keys (and HSM Transport Keys) are stored in the HSM.

Using a HSM in combination with the Vacman Controller guarantees that DIGIPASS Keys are never exposed on the Host computer.

1.2 Supported HSM Models

The Vacman Controller currently integrates with 5 different HSM models – **nCipher netHSM**, **Safenet ProtectServer Orange/Gold/External**, **Safenet LunaSA**, **Thales WebSentry** and **IBM ICSF**.

1.3 Glossary

Below are terms specific to the Vacman Controller for HSM integration.

Application Host	The back-end/host system where the user and transaction authentication takes place.
DIGIPASS Blob	See DIGIPASS Data.
DIGIPASS Data	DIGIPASS Data contains information about a particular DIGIPASS that is stored in the customer's application database and used in the Vacman Controller verification process. This includes DIGIPASS Key(s).
DIGIPASS Key	Cryptographic key stored both in the DIGIPASS and in the DIGIPASS Blob. It is used to generate One Time Passwords and Signatures using cryptographic algorithms.

DIGIPASS	DIGIPASS is a Hardware device or Software instance that is used to generate One Time Passwords and Signatures.
DPX file	A flat ASCII file that contains the DIGIPASS Data for each DIGIPASS in a particular batch. The DIGIPASS Keys inside this file are encrypted.
HSM	Hardware Security Module - a tamper-proof hardware device with cryptographic capabilities.
HSM Storage Key	Key stored in HSM and used long-term to encrypt/decrypt DIGIPASS Keys.
HSM Transport Key	Key stored in HSM and used to encrypt the DIGIPASS sensitive information inside the DPX file before the DPX file is sent from VASCO® to the customer. This requires that VASCO receives this HSM Key from the customer. The name of the HSM Transport Key, used to encrypt the DIGIPASS Keys during the transport, will be stored in the DPX file.
Integrator	The company/engineer that is responsible to integrate Vacman Controller into the application.
KCV	Key Check Value – created and stored in the DPX file to ensure the HSM Transport Key name has not been corrupted.
KEK	Key Encryption Key – used to encrypt the HSM Transport Key before transfer from customer to VASCO.
VC	Vacman Controller

1.4 Target Audience

This document is designed to assist VASCO customers and partners who are interested in integrating Vacman Controller into their application. It will allow them to make the necessary choices with regards to the integration of the Vacman Controller with a specific HSM.

2 HSM Integration Architecture

2.1 General

The Vacman Controller for HSM solution has been engineered so that it utilizes the general features of a HSM, as well as giving the customer the opportunity to integrate Vacman Controller into an existing or new application in a smooth way.

2.2 HSM Independence

Vacman Controller for HSM is provided with a level of HSM independence. That is to say, the VC library is a set of functions that give integrators simplified HSM support without the restriction of conforming to one particular HSM model. This will allow the simplest integration scenario – development will focus on application instead of HSM compliance.

On the Host, Vacman Controller does not perform any HSM Management processes (i.e. initialization/finalization, session management, communication). Integrators write this functionality separately from Vacman Controller, in line with the particular HSM in use.

Vacman Controller provides the Host-side generation/processing of the messages sent to/from the HSM. In this way, there is a complete separation of Vacman Controller and HSM Management processes.

For each Vacman Controller function performed inside the HSM, the Host-side Vacman Controller must perform:

- Generation of a HSM Command message to be sent from the Host to the HSM.
- Processing of the HSM Reply message that is sent back from the HSM.

Note that due to this separation of tasks, integrators must have knowledge of the operation and management of their HSM.

Note:

Exceptions are the Safenet LunaSA and the Thales WebSentry HSM. For these HSM models, the Vacman Controller communicates directly with the HSM through the PKCS#11 cryptographic token interface provided by the HSM.

2.3 Operating Modes

The customer must decide which operating mode to use. The choice may be limited if a particular HSM is already in place or a mainframe is used.

There are 3 operating modes:

1. **Data Encryption Only** – supported by IBM® ICSF® systems.
2. **Data Encryption and DES Operations** – supported by any PKCS#11 compliant HSM (Safenet Luna SA, Thales WebSentry).
3. **Data Encryption, DES Operations and VC Operations** – supported by nCipher netHSM and Safenet ProtectServer HSM's.

2.3.1 Data Encryption Only

In this operating mode, the HSM is simply used to store HSM Storage Keys and to encrypt and decrypt the DIGIPASS Keys. The DES operations and OTP validation will be processed by the Vacman Controller software that is running on the Application Host.

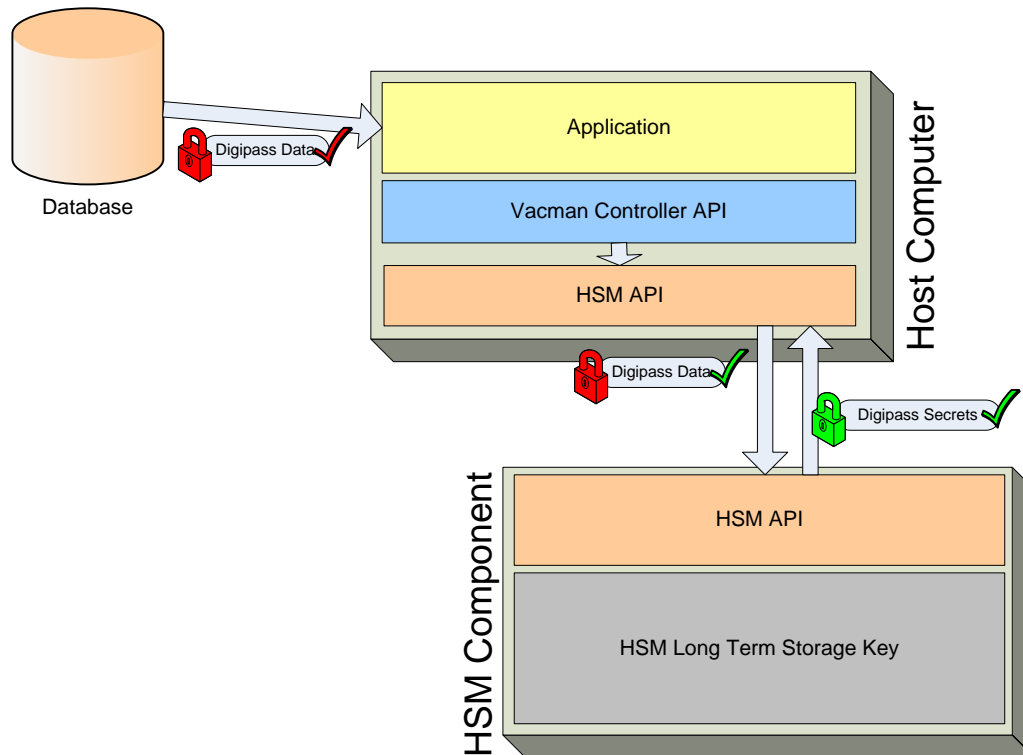


Figure 1 - Data Encryption Only architecture

With this mode we ensure that no one can use the DIGIPASS Data except the Application Host that has access to the HSM.

1. First the application retrieves the relevant DIGIPASS Data record from the database. Here, the DIGIPASS Data is secured by HSM Storage Key encryption (lock) and VC Software integrity (tick).
2. The Vacman Controller calls the HSM API with the encrypted DIGIPASS Data.
3. A HSM session is established and DIGIPASS Keys in the DIGIPASS Data are decrypted and returned.
4. The Vacman Controller uses the DIGIPASS Keys to perform 3DES and OTP validation.

This Vacman Controller for HSM Operating Mode is in use with the IBM® ICSF® system.

2.3.2 Data Encryption and DES Operations

With this operating mode, not only the secrets are in the HSM, but also the DES or other cryptographic processing is done inside the HSM itself.

The result of the cryptographic operation won't be accessible by any process outside the HSM. Output will be a cryptogram that the Vacman Controller will use to verify a One-Time Password or Signature code.

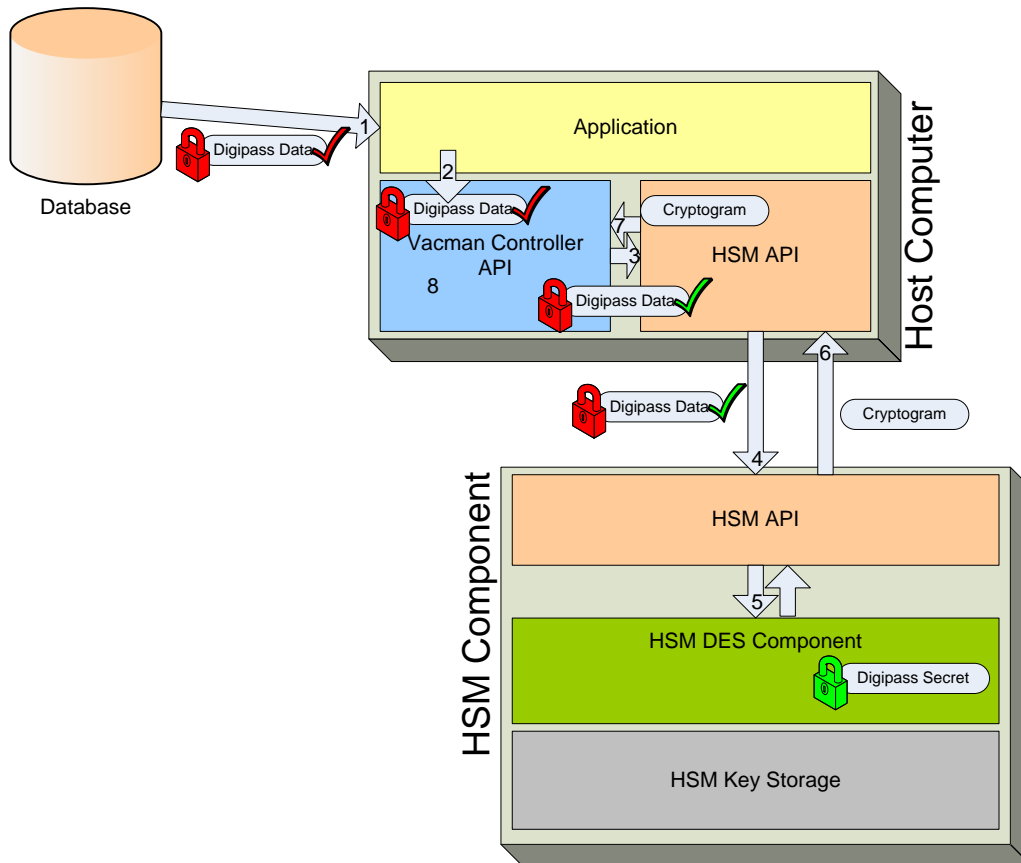


Figure 2 - Data Encryption and DES Operations architecture

1. First the application retrieves the relevant DIGIPASS Data record from the database. Here, the DIGIPASS Data is secured by HSM Storage Key encryption (lock) and VC Software integrity (tick).
2. The application calls a VC API, passing the DIGIPASS Data to the Vacman Controller (Software). The DIGIPASS Data is checked for integrity (Hash calculation).
3. The DIGIPASS Data, containing the encrypted DIGIPASS Key, is sent to HSM API along with HSM Storage Key name and the input of the DES.
4. The HSM API connects to the HSM and provides it with this information.
5. The DIGIPASS Key is then decrypted with the HSM Storage Key and the cryptographic (e.g. 3DES) calculation is performed by the HSM. (The DIGIPASS Key does not leave the HSM in clear).
6. The HSM returns the cryptogram (result of the cryptographic operation) to the Host side HSM API.

7. The HSM API provides the Vacman Controller (Software) with the Cryptogram.
8. The Vacman Controller (Software) will use this cryptogram to generate a Dynamic Password to match the one provided by the application (the end user).

With this operating mode, you ensure that the DIGIPASS Keys are never in clear on the Host side during an Authentication/Signature verification. This Vacman Controller for HSM Operating Mode can be applied on any PKCS#11 compliant HSM. It is in use with the Thales WebSentry and the Safenet LunaSA HSM. The main disadvantage of this approach is that the validation of a DIGIPASS password or signature may require several communications with the HSM, this for each DES/3DES calculation that needs to be performed.

2.3.3 Data Encryption, DES Operations and VC Operations

This mode will need a specific integration in the HSM – the Vacman Controller for HSM will reside in the HSM, depending on the vendor prerequisites.

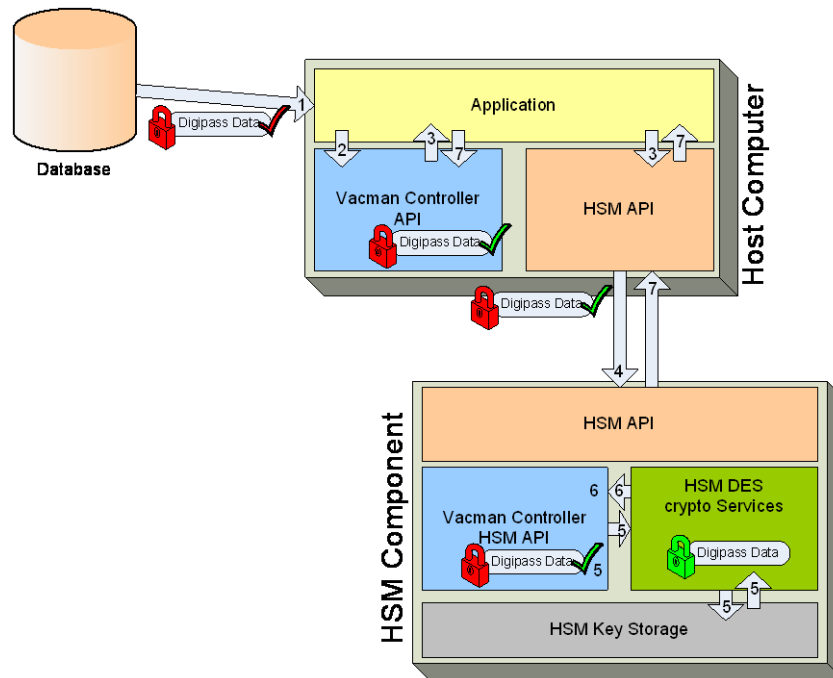


Figure 3 - Data Encryption, DES Operations and VC Operations

1. First the application retrieves the relevant DIGIPASS Data record from the database. Here, the DIGIPASS Data is secured by HSM Storage Key encryption (lock) and VC Software integrity (tick).
2. The DIGIPASS Data is passed to the Vacman Controller (Software part) to be checked for integrity (Hash calculation).
3. The DIGIPASS Data and Operation Request (e.g. OTP Verification) are formatted, serialized into a Command Message. This Command Message is sent to the HSM API (via the Application) along with credentials if needed.
4. The HSM API connects to the HSM and provides it with the Command Message.
5. The Requested Operation is then performed after another integrity check on the DIGIPASS Data.

6. The Vacman Controller HSM API has a handle to use the DIGIPASS Data (but not to read it) and perform OTP Verification.
7. The Vacman Controller for HSM API creates a Reply message. This includes updated DIGIPASS Data and a Operation Success Status code. The reply is provided back to the Software Vacman Controller via the application and the HSM API's.
8. The Software Vacman Controller uses the handle to DIGIPASS Data to update it and pass it to the application (that will store it in the database).

With this operating mode, you ensure that the DIGIPASS Keys are never in clear on the Host side during an Authentication/Signature verification, but also that no process on the Application Host has access to the cryptographic operation results. The HSM acts as a 'black box' where the critical authentication process is heavily protected. Already, nCipher netHSM and Safenet ProtectServer are supported.

3 Vacman Controller for HSM – Additional Features

3.1 DPX Double Encryption

The DPX file generation process can be modified so that the DIGIPASS Keys are first encrypted by a HSM Transport key. The encrypted DIGIPASS Keys are embedded in the DPX file and additionally encrypted in the standard DPX way, using a software security module (Software level Transport Key).

As a result, the DIGIPASS Keys in the DPX are double encrypted and the remaining data single encrypted.

The purpose of this is to assure the DIGIPASS Keys on the Application Host are continuously secured by HSM-level encryption (Transport or Storage Key) - the DIGIPASS Keys will never be exposed during this process.

When importing the DPX file, the DPX to DIGIPASS Data conversion will be carried out in the normal way, in software. The (software-decrypted) DIGIPASS Data would then contain DIGIPASS keys encrypted under a HSM Transport key.

The AAL2MigrateBlobHSM function is used to decrypt the DIGIPASS Keys using the HSM Transport key and re-encrypt them under the HSM Storage key.

The final DIGIPASS Data is encrypted at the software level. This creates a DIGIPASS Data entirely compatible with the existing VC for HSM integration, using exactly the same double-encryption format for the DIGIPASS Keys inside the DIGIPASS Data.

Modified Import Process

For the Vacman Controller Integrator, the DPX import process is slightly modified. It is now necessary to add an extra VC API call after retrieving a DIGIPASS Blob – AAL2MigrateBlobHSM. Sample pseudo-code follows below.

```

For each different application name in DPX File Do
  AAL2DPXInit()
  For each DIGIPASS in DPX file with current application name Do
    AAL2DPXGetToken()
    AAL2MigrateBlobHSM()
    Store DIGIPASS Blob in Application Database
  EndFor
  AAL2DPXClose()
EndFor

```

Prior to any password or signature validation it is required that you migrate the DIGIPASS Data encryption in this way. The Vacman Controller API Authentication and Management functions will reject a DIGIPASS Blob that is encrypted with a HSM Transport key.

HSM Transport Key Import

The HSM Transport Key resides on the customer's HSM. However it must be sent to VASCO before any DPX double encryption process can commence. Standard procedures for this Key transferal are available – the customer must fill out a Request for import of HSM-level DPX Transport Key for Double DPX Encryption form (more detailed information is available upon request).

The actual transfer involves use of a Key Encrypting Key (KEK) – this KEK has previously been created and shared by the customer and VASCO. The KEK is used to encrypt the HSM Transport Key so as to provide a secure means of transferring it.

3.2 HSM Storage Key Migration

To update the HSM Storage Key that is used to encrypt the DIGIPASS Blobs, there is the AAL2MigrateBlobHSM function.

With this function you pass the old and new HSM Storage Key names (or old and new HSM Storage Key ID's). The DIGIPASS Blob now has HSM-level encryption using the new HSM Storage Key. From then on you must pass the new HSM Storage Key to the verification process and other VC functions.

3.3 HSM Load Balancing and Fail Over

The double encrypted DIGIPASS Blobs will be rendered useless if the HSM Storage Key, used to encrypt them, is lost or corrupted, or in the event that the HSM fails. In this case, there are 2 options recommended:

- Provide a backup HSM containing the same HSM Storage Keys.
- Provide secure backup media that will enable you to restore the HSM Storage Keys.

3.4 HSM Key Names and Identifiers

Vacman Controller uses specific names or ID's for referencing the HSM Storage Key and HSM Transport Key.

Note that you may use any of these names / ID's for either HSM Storage Key or HSM Transport Key.

HSM Key Name	HSM Key ID
vascoStorageKey	0
vascoTransportKey	0x7FFFFFFF
vasco1	1
vasco2	2
vasco3	3
vasco4	4
vasco5	5
vasco6	6
vasco7	7
vasco8	8
vasco9	9

4 Additional information

Please refer to the following White Paper - Vacman Controller Integration.

5 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.