Thank you for joining.

Your Webinar will begin shortly.

utimaco®

Protecting Data and Critical Infrastructure with Utimaco Hardware Security Modules (HSMs)

General Purpose HSMs Webinar

Richard Williamson - Director of Technical Services
and Randy Petersen - Senior Systems Engineer

utimaco®

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- Use Cases
- Integrations
- An overview of the state of Crypto
- Decision Tree
- Things to Think About

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- **Hardware Security Modules - Form factors, Use**
- Regulatory Bodies - Standards and Requirements
- Use Cases
- Integrations
- An overview of the state of Crypto
- Decision Tree
- Things to Think About

Fun Fact:

There is no official pronunciation of the company name

Hardware Security Modules

Form factors

- Secure Execution Environment
  - Like: Intel SGX, ARM TrustZone
  - In-Chip area with “intrinsic” hardware protections
  - Primary Purpose: application (data-in-use) protection
  - Roll Your Own Security

- Trusted Platform Module
- Hardware Security Module

Hardware Security Modules

Form factors

- Secure Execution Environment

- Trusted Platform Module
  - Various manufacturers
  - Discrete chip on motherboard
  - Primary Purpose: system (data in motion, data at rest)
  - Limited storage, fixed algorithm choice

- Hardware Security Module

Hardware Security Modules

Form factors

- Secure Execution Environment
- Trusted Platform Module

- Hardware Security Module
  - Various manufacturers
  - Discrete computer that you plug into another computer (i.e., PCIe card)
  - Sold as an appliance (plumbing for the PCIe card)
  - Primary Purpose: System-of-systems protection for data at rest, in motion and in use.

Utimaco CryptoServer

Se12, Se52, Se500, & Se1500

CSe10 & CSe100

CryptoServer LAN Appliance

Each of the (6) PCIe card HSM devices is also available in a CryptoServer LAN appliance form factor.

Hardware Security Modules

Purpose

- The PURPOSE of an HSM is to PROTECT THE KEYS
  - At rest, and in use. Ideally, they will never be ‘in motion’ except in limited, specific cases
  - The goal is to not sacrifice security for performance
  - The goal is to not sacrifice security for convenience of use

- Which keys?
  - Symmetric Keys (AES)
  - Asymmetric Keys (RSA, ECC)
  - Any arbitrary secret

Hardware Security Modules

Terms - API vs SDK programming

- Programmable, General Purpose HSMs
- APIs (host-side application programming interfaces/libraries)
  - PKCS#11
  - CNG (CSP, CNG, KSP, EKM, …)
  - JCE/JCA
  - The vendor's own API
- SDK (HSM-resident code, i.e. within the security envelope of the physical HSM)
  - Customized anything
  - Utimaco's SDK allows you to turn a multi-step process into an "atomic", single call to the HSM
    - Performance Gain: Reduces calls to the CryptoServer, thus avoiding network latency
  - Utimaco's SDK allows you to write your own Crypto mechanisms, algorithms, etc..
    - Avoid Obsolescence: Processes, algorithms etc.. can evolve, you are not limited to what is shipped with the hardware (i.e., TPMs)
  - Performance improvements and avoiding obsolescence are orthogonal to each other, both can be achieved within a single custom module

Hardware Security Modules

Terms - Auth'N and Auth'Z

- MFA - Multi-Factor Authentication
  - "Something you have, something you know"
  - Using a PIN-protected smart card or similar to log into an HSM, for admin or use
- 4-eyes - More than one person needed to authenticate the session
  - Defeats the single insider threat
  - Also may be known as M-of-N (3 of 5, 2 of 7 etc.)
- MFA and 4-eyes should not be mutually antagonistic, a well-designed system allows you to use either or both, depending on your security profile and threat level at each access level
  - 4-eyes HSM Admins with smart cards, or a single 'cryptographic user' with an HMAC-encrypted password with access to a single key and its use
  - Or anything in between
- Session-based Authorization - what a session is allowed to do
  - Based on who (plural, 4-eyes) is logged into the session

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- **Regulatory Bodies - Standards and Requirements**
- Use Cases
- Integrations
- An overview of the state of Crypto
- Decision Tree
- Things to Think About

Fun Fact:

There is no “L” in Utimaco and only one “T” (so, no, we are not ‘Ultimatico’)

Regulatory Bodies and Certification

Requirements

- A specific standard may be demanded by a government,
  - FIPS 140-2
  - Common Criteria EAL4+
  - Regional homologation requirements (technically, FIPS 140-2 is a regional homologation)
- A specific standard may be demanded by a trade body,
  - PCI HSM (Payments)
- A specific standard may be demanded by your CISO.
  - Any of the above
  - A “vendor independent” API (PKCS#11, CNG, JCE)

Regulatory Bodies

Standards

- FIPS 140-2
  - US NIST
  - To be FIPS 140-2 certified, is to be certified to one of four levels
    - Level 1: *Approved Algorithms only*
    - Level 2: *Tamper Evident*
    - Level 3: *Tamper Resistant - Can only be achieved with Hardware*
    - Level 4: *Tamper Reactive - Can only be achieved with Hardware*

    - Note, if certified to any level, this requires you also be subject to the restrictions of the lower levels. You cannot be Level 3 certified, unless you are also running only approved algorithms and are tamper evident.

Regulatory Bodies

Standards

- Common Criteria
  - EU requirements
  - There are different “protection profiles” for different use cases
    - eIDAS: “Cryptographic Module for Trust Services” EN 419221-5
  - There are different EAL levels (Evaluation Assurance Levels)
    - EN 419221-5 requires EAL4+ (Evaluation Assurance Level 4+)

Certification Requirements

Issues

- Most of the previous are mutually incompatible.
  - You can not be both PCI-HSM certified, and run in FIPS mode
    - Among other reasons, PCI-HSM *requires* DEA (DES), but DEA is not an approved algorithm.
  - If you are running custom (SDK) code, you can't run in *any* certified mode out of the box
    - Requires a "delta" cert, or a complete recertification, depending on the standards body
- Many standards are behind the times
  - PCI-HSM still uses 3DES, AES is not yet mandated
  - FIPS approved RSA only supports key sizes up to 3096
- Many standards are, conceptually, locked to a certain mindset
  - PKCS#11 was originally for Smart Cards
  - Extended to use in HSMs, but still treated like using Card Readers and Tokens
  - Supports only the "Security Officer" and "PKCS#11 User"
  - 4-Eyes or MFA have to be implemented 'inside' the parent concept
    - So each vendor's "standards-based PKCS#11 provider" will have different ways to do it
    - Negates the whole 'vendor independence' goal.

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- **Use Cases**
- Integrations
- An overview of the state of Crypto
- Decision Tree
- Things to Think About

Fun Fact:

CryptoServer HSMs are manufactured in a secure facility, in Germany

How an HSM is used

(What is meant by) Data at Rest, in Motion, and in Use

- Data at Rest - Data is encrypted when stored, and decrypted for use
  - Used by Databases for “TDE”
    - Table Data Encryption
  - The database derives a symmetric key, which it uses locally to encrypt the table data. The master key for this is stored on the HSM.
- Data in motion - Data moving between systems or applications
- Data in use - Data in memory or in registers

How an HSM is used

(What is meant by) Data at Rest, in Motion, and in Use

- Data at Rest - Data is encrypted when stored, and decrypted for use
- Data in motion - Data moving between systems or applications
  - Used for ssh and https connections
  - Diffie-Helmann (DH) Key Exchange to set up a shared symmetric key, the asymmetric keys remain on the HSM.
- Data in use - Data in memory or in registers

How an HSM is used

(What is meant by) Data at Rest, in Motion, and in Use

- Data at Rest - Data is encrypted when stored, and decrypted for use
- Data in motion - Data moving between systems or applications
- Data in use - Data in memory or in registers
  - Data is sent to where the keys are (on the HSM) for processing
  - Prevents a well-timed core dump, or JTAG investigation of the data and keys at runtime/in memory/in registers
  - Ask your vendor if their HSMs have a JTAG port. He/she should laugh at you. If no laughter is forthcoming, either they have no sense of humor, or you should be very, very scared.

Use Cases

Code Signing

- Modern operating systems insist that the application be signed
  - Where is that key that is used to sign the application or shared libraries?
    - On an HSM
- Different operating systems, different environments have different tools
  - Jarsigner
  - Authenticode
  - etc.
  - *The keys are stored on the HSM, generally a hash of the object is sent to the HSM, a signature is returned, the signature is appended to the object as its signature. The OS/environment can repeat these steps, and verify the signature using the signer's public key.*

Use Cases

Web-Server Security

- All modern web servers
  - Apache2
  - Tomcat
  - Nginx
  - IIS
  - …
- Can act as HTTP and HTTPS servers.
  - HTTPS requires a DH key exchange with the remote browser
  - The server’s key should be stored in an HSM
  - Prevent man-in-the-middle attacks

Use Cases

Securing the CA

- The Root CA (or, more likely, via CSR to a public CA)
  - Generate the key for your root Certificate Authority on an HSM
  - Generate a “self-signed certificate” for the key, using the HSM
  - Or, Generate a CSR for the key using the HSM, then submit the CSR to the public CA for signing. Then use your HSM-resident key to bless your own subordinate (issuing) CAs for use

- It depends on who your systems are talking to --
  - A root CA is useful for when the systems will be talking to each other in the (private) system
  - A CSR signed by a public CA is useful for when the systems will be talking to other parties/the public
  - The Public CA acts as an unbiased, trusted third party

Use Cases

Database Encryption

- Oracle TDE
- Microsoft SQL Server

- Libraries provide access from the database engine to the HSM
- HSM is used primarily for storage of master keys, derived keys are used as needed to encrypt/decrypt the data. These steps are done on the host, because of performance

Use Cases

And everywhere else

- Others.
  - In general: Does the Software support the use of an HSM for its cryptographic requirements?
  - In general: Does the HSM support the required algorithms the software needs?
  - Then: There is no reason not to use an HSM to support the software
  - However:
    - The software may not be written such that it can use an HSM
      - We have a simulator! Try it an see!
    - The software may use an API that the HSM does not provide
      - i.e., BouncyCastle isn't abstracted, so can't use an HSM
    - The software may require a mechanism/algorithm that the HSM does not provide out of the box
      - We have an SDK, you can add it yourself!
  - "HSMs are too expensive"
    - And exposing 400m public records isn't?
  - "HSMs are too slow"
    - You don't want an HSM, you want a DH accelerator, sacrificing security for performance

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- Use Cases
- **Integrations**
- An overview of the state of Crypto
- Decision Tree
- Things to Think About

Fun Fact:

Using a search engine on “HSM” generally gets you information on “High School Musical”

Integrations

What is an integration?

- Many applications exist that are designed to protect your systems
- Many ARE written to leverage HSMs where needed

- Hopefully, nothing in the software should prevent, or preclude the use of an HSM
- Hopefully, nothing in the HSM should prevent, or preclude the use of your software

- In general, applications will be written to use one of the standards-based APIs
  - PKCS#11, CNG, JCE

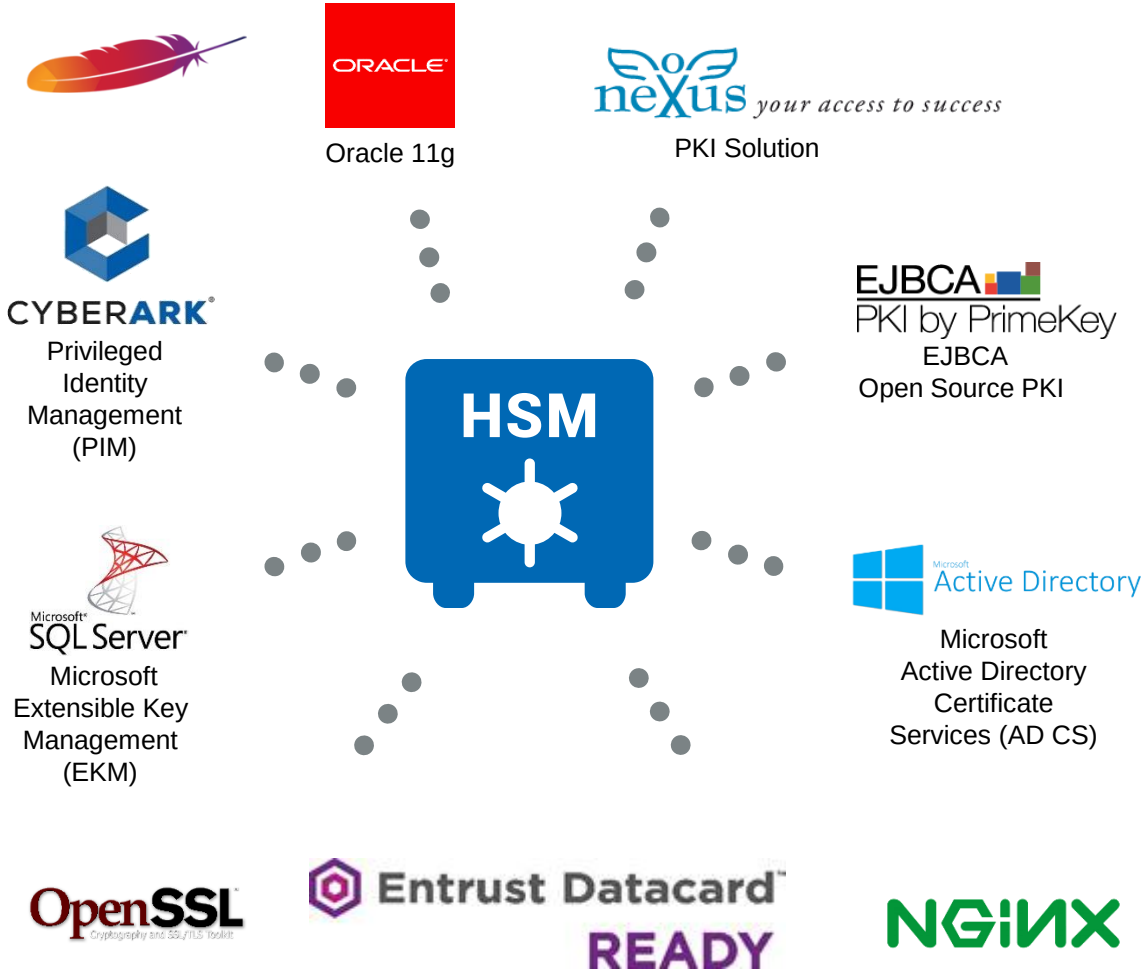
- In this case, there is a good chance that it will just work
- Sometimes, the software makes assumptions about how the standard is implemented
  - *These assumptions can differ from reality*

Integrations

The good news! You don’t have to do everything from scratch

- The key is:
  - In general, they will be written to use one of the standards-based APIs (PKCS#11, CNG, JCE)
- Utimaco does provide integration guides for many of these targets:
  - HTTPS
  - IAM
  - TDE
  - PKI
  - Code Signing
  - OpenSSL
  - etc.
- https://support.hsm.utimaco.com/support/documentation/integration-guides

Selected integrations



Integrations

What is an IG?

- Integration Guides are written from the viewpoint that:
  - You already know how to use the software application
  - You are new to HSMs in general/Utimaco CryptoServers in general
  - You just need to know how to configure/administrate the CryptoServers, so the software can use it.

  - The IGs provide the steps to get there.

Integrations

What if there is no IG?

- Contact the software vendor, and ask if it supports the use of an HSM
  - Helps if we know what standard API they target (PKCS#11, CNG, JCE?)
  - If so:
- Utimaco technical staff will try to contact their technical staff
  - Exchange of test/debug environments (Did I mention, our Simulator?)
  - Exchange of know-how
  - A new IG is born

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- Use Cases
- Integrations
- **An overview of the state of Crypto**
- Decision Tree
- Things to Think About

Fun Fact:

The CryptoServer Simulator can be used to step-through debug your custom SDK modules.

Cryptography concepts

Symmetric / Asymmetric

- Symmetric
  - Algorithms like DES and AES
  - Mechanisms like CBC and ECB to GCM
  - Primary Use: encrypt/decrypt
  - Symmetric: Same key is used for both sides of an operation (encrypt/decrypt)
  - Fast

- Asymmetric
  - Algorithms like RSA and ECC
  - Primary Use: sign/verify, key agreement
  - Asymmetric: Different keys (paired ‘public’ and ‘private’ keys) that are mathematically linked
    - Private key used to Sign, Public key used to Verify
    - Public Key used to Encrypt, Private key used to Decrypt
  - On average, much more compute intensive than Symmetric algorithms

Post-Quantum Cryptography

Concepts

- A ‘classical’ computer uses bits, each with a state of On or Off (1 or 0).
- A ‘quantum’ computer uses qubits, quantum bits, each with a state ranging from On to Off (1.0 to 0.0)
  - Actually, they represent *all* states between 1.0 and 0.0, at the same time
  - (Physicists in the audience are now quibbling with this definition)
- Classical computers are used to IMPLEMENT crypto.
- Quantum computers are going to be used to BREAK crypto.
  - Shor (factors an integer N into its prime factors, ie, the math behind RSA)
  - Grover (useful for searching large data fields, ie to find a symmetric key)

Post-Quantum Cryptography

Concepts

- Shor (factors an integer N into its prime factors, ie, the math behind RSA)
  - Doubling the key length only provides a marginal increase in security
    - To Shor's algorithm, an 8K RSA key is only marginally more secure than a 4K RSA key
    - Shorter keys are effectively useless. ECC keys are measured in 100s of bits, not thousands, so are effectively transparent (assuming a sufficient number of qubits)
- Grover (useful for searching large data fields, ie to find a symmetric key)
  - To Grover's algorithm, a Symmetric key is half as secure.
    - Fix: Double the size of your symmetric key
      - An AES 256bit key is still effectively opaque, as we can't break AES 128bit keys.
- Fix: Use a quantum-safe algorithm
  - Picnic, New Hope, Dilithium, Kyber, …
  - Many different algorithms have been put before NIST, who will bless the lucky few with the label "Approved Algorithm", making them available for use in FIPS 140-2 Level 1 and higher systems.
  - Q-Safe is already available, a plug-in module to the standard GP HSM line (CryptoServer)
    - Currently provides as a demo, Dilithium, Kyber, and will provide XMSS and XMSS-MT

Post-Quantum Cryptography

Things to remember about PQC

- “Stable” is the primary thing. The qubit has to be available and steady, long enough for the answer to “fall out” of the computation.
- Adding a single qubit doubles the ‘strength’ (computing power) of a quantum computer.
- Experts agree that a single quantum computer with enough stable qubits (~56 qubits) will exist in the next 10 years, sufficient to break current asymmetric key cryptography (RSA, ECC)
- Most state-level actors are currently harvesting DH key exchanges and the resulting data streams. Once they have sufficient stable qubits in a single computer, they will:
  - Open the key exchange of an interesting stream to get the values used to generate the shared secret,
  - Recompute the initial shared secret, and then decrypt the data stream.
  - Profit!
- You have been warned.

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- Use Cases
- Integrations
- An overview of the state of Crypto
- **Decision Tree**
- Things to Think About

Fun Fact:

I have a spreadsheet of all the street addresses I've had. There are 54 rows.

Decision Tree - Standard Crypto

utimaco®

Your use case does not require PQC or access to custom, HSM-resident code

- Does your primary use require FIPS 140-2 Level 4 for physical security?
  - Standard performance: CSe-10
  - High performance: CSe-100

- No, Level 3 is sufficient.
  - Does your primary use case require RSA or ECC?
    - Standard Performance: Se-500
    - High performance: Se-1500

- Ok, you’re focused on Symmetric cryptography
  - Standard Performance: Se-12
  - High Performance: Se-52

- No additional licenses required (no counted users, connections, applications)

Development

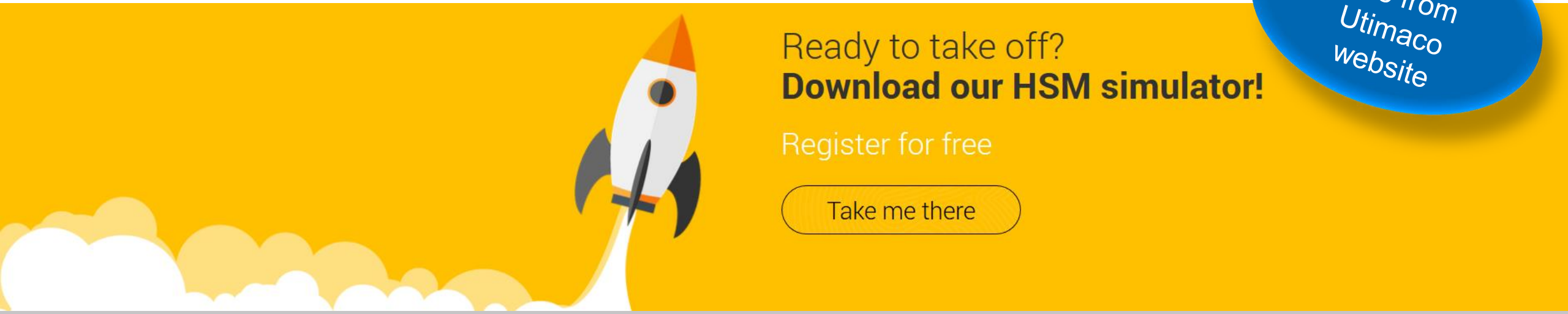
Sim -> Steel -> Cloud

- Download the Simulator:
  - Administrate the simulator (users, etc.)
  - Configure your application to use it: Device = 3001@127.0.0.1
  - Develop your application against the Simulator
- Migrate to Hardware:
  - Administrate the hardware (users, etc.)
  - Change the configuration file: Device = 288@10.0.1.200 288@10.0.1.201 …
  - (N-count devices running as a cluster for HA, FT, Performance, etc.)
  - Rerun your application
    - Yep. That's all.
- Migrate to Cloud
  - Administrate the Cloud HSMs (users, etc.)
  - Change the configuration file: Device = 288@199.199.199.201 …
  - Rerun your application
    - Yep. That's all.

Free HSM simulator

- Fully functional software simulator for Windows and Linux
  - HSM administration, user authentication, key management, cryptography, etc..

- Ideal for
  - Product evaluation
  - Dry-run before setup of production HSM
  - Integration testing
  - Training



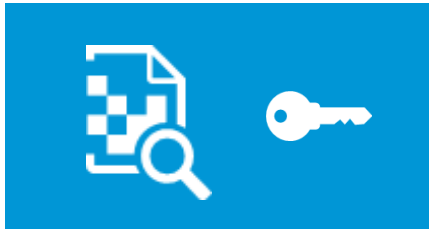
Decision Tree - Notes

Additional cost items

- Q-Safe (NIST-approved post-quantum algorithms)
- SDK (for writing your own applications to run directly within the security envelope)
- Training
- Professional Services
- Maintenance/Support

Utimaco Solutions

Portfolio



Utimaco CryptoServer
General Purpose HSM – Provides Root of Trust

Enterprise Secure Key Manager (ESKM)
Creates, serves, and protects encryption keys for enterprise key management

Utimaco Atalla Hardware Security Module (HSM)
Also known as Atalla Payments HSM – leading product in payments security

Utimaco Cloud Encryption (UCE)
Bring your own keys to the Cloud

Agenda

Utimaco General Purpose Hardware Security Modules (GP HSM)

- Hardware Security Modules - Form factors, Use
- Regulatory Bodies - Standards and Requirements
- Use Cases
- Integrations
- An overview of the state of Crypto
- Decision Tree
- **Questions?**

Fun Fact:

This is the penultimate slide.

How do we lead?

Innovation

46 Patents

Creative engineers delivering security inventions and driving security thinking

Rock-solid security

FIPS 140-2 validated Level 2, 3, and 4

Our Key Management Solutions are built for the highest standards

Trusted name since 1972

$ Trillions

Utimaco Atalla secures 1 in 3 card transactions; also processes billions of card transactions annually

Q&A – send to all panelists

Presenters: Richard Williamson and Randy Petersen

Email: academy@utimaco.com

…and we invented security that you can take for granted!!!