IT-Sicherheit & Cloud

Elastic Key Management and Homomorphic Encryption

ISACA Germany Chapter
Meet & Explore
Deutsche Telekom AG
Bonn, 3.9.2015

Alexander W. Koehler
Dipl.Math. CISSP, CCSK, CSP
Elastic Key Management and Homomorphic Encryption

• Agenda
  – Scope and Expectations
  – Specifications and Specifics
  – Status
  – Scope of Requirements
  – Homomorphic Encryption
  – How it works
  – What it means to Go4Cloud
  – Implementation Details
  – Alternative Approaches
  – Take Aways
  – Credit Goes to

• Q & A
Scope and Expectations

– Controls
  • „On-Premise“ controls are not suitable for „Cloud“
  • Controls designed for Cloud outperform „On Premise“

– Elastic Key Management and Homomorphic Encryption (EKM&HE) is the key enabling technology for Cloud
  • EKM&HE enables customers to Go4Cloud: TRUST!

– EKM&HE is a breakthrough in technology as well as operational service
Specifications and Specifics

- **Players:**
  - Customer (CUS)
  - Cloud Service Provider (CSP)
  - Security Services Provider (SSP)
  - Authorities (GOV)
- **Public Cloud (AWS)**
- **Data processing in the Cloud**
- **Elasticity**
- **Multi-Tenant**
- **C I A:**
  - A -> CSP
  - I -> CSP and CUS
  - C -> CUS
Status

• **Inhibitors** of „Go4Cloud“:
  – No. 1: Data Protection and Compliance: 60%->75%
  – No. 2: Information Security 49% -> 67%

• **Source**: Study PricewaterhouseCoopers, 03/2013: „Cloud Computing – Evolution in the Cloud“

• **Conclusion**: Current cloud offers as well as technologies are not yet there to create the main prerequisite for Go4Cloud: **Trust**
Scope of Requirements

- CUS shall have unique control: **Trust**
- Highest level of data entities protection
  - Encryption
    - Keys
      - Key Management
        » Key protection
        Auguste Kerckhoffs, Nuth, Netherlands, 1835 – 1903
- Elasticity applicable to all components
Homomorphic Encryption

• Definition:
Homomorphic Encryption is a form of encryption which allow specific types of computations to be carried out on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations on the plaintext.

• Fully Homomorphic

• Partially Homomorphic

\[
\begin{align*}
\text{Encr}(a) + \text{Encr}(b) &= \text{Encr}(a+b) \\
\text{Decr}(&\text{Encr}(a+b)) &= a + b \\
\text{Encr}(a) \times \text{Encr}(b) &= \text{Encr}(a \times b) \\
\text{Decr}(&\text{Encr}(a \times b)) &= a \times b
\end{align*}
\]
Homomorphic Encryption

- Data processing is done on ciphertext
  - Data processing in the Cloud
    - **without presence of keys in plain** in the Cloud
  - Data stays under the control of the CUS
Homomorphic Encryption

• Homomorphic data encryption downside
  – Requirements for processing power cannot be fulfilled with currently available algorithms

• Bruce Schneier: “Performing a web search with encrypted keywords – a perfectly reasonable and simple application of this algorithm – would increase the amount of computing time by about a trillion.”
Homomorphic Encryption

• Performance impact:
  – Encryption
    • of data
    • of keys
    • of Master Key

• Implementation: Homomorphic encryption of the Master Key
  • Master Key in the Cloud is encrypted
    • At rest
    • During operation
How it works - 1

• An application has done data processing and will store the resulting data:
  
  Protected:

  • A Virtual Appliance (PVA) is hooked between App and Storage: every data block goes through the PVA
How it works - 2

- The encryption key is split in 2 key halves

- Master Key: per application or per project
- 2nd key half: generated for every data storage object
How it works - 3

- There are keys in the CUS Cloud environment
- Auguste Kerkhoffs’ principle
- All keys are stored encryptedly
- **Master Keys** stay encrypted, **incl. during processing**
- 2nd key halves are randomly generated keys **per data object**
How it works - 4

• Homomorphic Split Key Encryption allows the described proceeding
What it means to Go4Cloud

- **Trust** and Compliance
  - Master Keys are under full control of the CUS
- **Unlimited benefits** of Cloud
  - CUS is not obliged to build his own key management infrastructure
  - CUS is not obliged to acquire key management appliances in data centers and manage them
  - Neither impact on performance nor provisioning
  - Deployable in minutes
What it means to Go4Cloud

- **Resilience**
  - Worst case: Breach into CUS VE
    - Exposure of only the one data object in processing at a certain moment of time
  - The breach of a single object cannot lead to a breach of the entire system
  - This is a level of protection that even on-premise encryption solutions cannot offer
    - Technology breakthrough
Implementation Details

- Encryption of data
  - AES 256-bit, CBC
  - Encrypted Salt-Sector Initialization Vector (ESSIV)
- Encryption of keys
  - Homomorphic Key Management (HKM) protocol: ElGamal k= 2048, SSL

- Porticor Virtual Key Management (PVKM) service - Security Services Provider
- Porticor Virtual Appliance (PVA) – in CUS environment
- Invented, patented and software developed by Porticor Ltd., Hod Hasharon, Israel and Campbell, USA
  - Porticor® Virtual Private Data™ System
  - www.porticor.com
Alternative Approaches

• Key management on-premise (PKI,...) or Tokenization
  – Requires additional infrastructures on-premise

• Key management in the Cloud
  – Software only
    • Low level of protection, no real benefit
  – HSMs, provisioned by CUS
    • No protection of data in processing
    • Requires acquisition of hardware in data center
  – HSM as a service, provisioned by CSP
    • No protection of data in processing
Credit Goes to

- Gilad Parann-Nissany, Founder, CEO
- Ariel Dan, Co-Founder, EVP Sales and Marketing
- Prof. Reimer, ISSE, Teletrust
- Prof. Pohlmann, ISSE, Teletrust
Thanks for Your Attention

Alexander W. Koehler
Contact:
Website: www.ict5.com
Blog: www.ict5.de/aktuell
Email: awkoehler@ict5.com
Phone: +49 (69) 6607 6603
Take Aways

• Elastic Key Management and Homomorphic Encryption (EKM&HE)
  – The key prerequisite technology to Go4Cloud: Trust
  – Will unlock resistance of CUS to Go4Cloud: Trust
  – Does not cut into Cloud inherent benefits
  – Does not expose keys in plain during processing
  – Migrates Kerckhoff’s principle in the new world of Cloud