Technical Level: Advanced

DORA ICT Encryption

How My Own Servers Could Do Better

Santosh Pandit

London, 6 November 2024

Contents

Disclaimer

Please note that I am agnostic when it comes to vendors and their products. For example, I have used more than 20 cloud service providers, and each one of them has its own pluses and minuses. Therefore, if I have cited any names as examples, it is only because many of my readers are using those vendors and their products.

Just because I use a product does not mean it is suitable for you. Likewise, if I discontinue using a product, it may still be perfectly fit for your use case.

All analysis and views are strictly personal, and any mistakes are solely mine.

Context

I enjoy experimenting with applied cryptography in my private laboratory and am proud that the cyber posture of my server easily surpasses that of Fortune 100 companies. Of course, my own servers are not required to comply with DORA.

In a hypothetical scenario, if I had to implement DORA in my laboratory both in letter and spirit, I would find the ICT Section 4 (Encryption and Cryptography) extremely challenging.

But every challenge is an opportunity. In this article, I will share a spectrum of potential approaches to addressing encryption and cryptography needs, along with my personal views on the nuances involved. Please note that I believe in a **forward-looking approach to managing technology risk**. Therefore, the investment I would make must make business sense, and DORA compatibility is simply a bonus.

Here is the exam question. How do we support the strategic objective of cyber resilience by making a smart investment in encryption and cryptography?

The Spectrum[1]

I would suggest the following ten topics need attention in our strategic planning and technology choices. Although overlaps are inevitable, I have tried to minimise them.

Of these ten topics, those marked in **green** are prescriptive requirements under DORA ICT 1774. Those in **blue** come across as optional requirements. Those in **red** are my chosen additions to complete the holistic picture, where I believe the DORA may fall short.

1. **Policy & Cryptographic Controls**
2. **Encryption for Data at Rest**
3. **Encryption for Data in Transit**
4. **Confidential Computing / Encryption for Data in Use**
5. **Key Management (& HSM)**
6. **Certificate Management**
7. **Cryptographic Updates & Monitoring**
8. **Compliance & Audit Tools**
9. **Key Rotation & Lifecycle**
10. **Backup Encryption**

Allow me to explain the items in **blue** and **red**.

Confidential Computing / Encryption for Data in Use

Please note that DORA ICT 1774 does not explicitly use the term “confidential computing.”

Confidential computing refers to a technology that protects data in use by isolating sensitive computations within secure, hardware-based environments known as Trusted Execution Environments (TEEs). This approach ensures that data is encrypted and remains confidential, even while being processed, thereby enhancing data privacy and security in cloud computing and multi-party environments.

In my view, because consultants and vendors tend to throw everything into the mix, it becomes difficult to set “confidential computing” as a precise target. DORA instead references the more specific concept of encryption for “data-in-use.” Article 6.2(b) states: “the encryption of data in use, where necessary.”

In my opinion, encrypting data in use within cloud environments is essential to guard against host-level attacks targeting containerised applications. Although it cannot be proven, we should be safer than sorry in protecting the “data in use” within our infrastructure against sophisticated actors such as the Equation Group, Fancy Bear (APT28), DarkSide, and Charming Kitten (APT35).

In the table further below, I have referred to commonly used technology for confidential computing, but do not forget that in most cases, you would use a combination of hardware and OS settings. For physical devices, BIOS settings must be enabled, and for all devices, encryption should be activated at the OS boot stage, as illustrated in the GRUB configuration example below:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash kvm_amd.sev=1"
```

Of course, you already knew that.

[1] I will keep the topic of ‘quantum resistant computing’ outside of the scope of this article.

Hardware Security Modules (HSM)

Firstly, I recommend reading Articles 6 and 7 of DORA ICT 1774, as they are necessary for informed decision-making. Article 6 allows for compensating controls and permits documentation when certain measures are "not possible." However, Article 7 consistently uses the term "shall," which indicates mandatory actions.

My interpretation of these two ICT articles is that single-tenant (i.e., dedicated rather than shared) servers are required, and the use of HSMs becomes inevitable when high-risk factors are identified in an ICT risk assessment (as per DORA Article 6).

Your interpretation may differ, and I would be interested to hear more about the technology stack you would consider for critical applications.

Key Rotation

Key rotation is the process of regularly changing cryptographic keys used for encryption, decryption, or signing data to enhance security. Its importance lies in mitigating the risk of key compromise; if a key is exposed or stolen, rotating it ensures that any unauthorised access is limited to the time the compromised key was in use. **Table A** contains my preferred rotation frequency, your choices may vary. Be aware of the 'harvest now, decrypt later' risk.

Table A – Preferred Key Rotation Frequency

Type of Key	Recommended Rotation Frequency
Encryption Keys	Every 1 year (careful about backups)
Signing Keys	Every 1 year
Session Keys	After each session
API Keys	Every 6 months (careful about applications, particularly third-party access)
SSH Keys	Every 1 year (or longer if you use a passphrase)
TLS/SSL Certificates	Every 1 year (note this is getting shorter)
Encryption Keys for Backups	Every 1 year
Database Encryption Keys	Every 1 year (careful about applications)
HSM Keys	Every 1 year (if possible)

Backup Encryption

DORA prominently mentions backups and encryption in different places, so allow me to focus on a nuance that may have been overlooked. When we need to use the backup, we could find ourselves in a scenario where the traditional storage of encryption keys may not be available. Without those keys, you will not be able to restore from your backup.

When encryption keys are lost or destroyed, we face a situation similar to that of a destructive ransomware attack. I would therefore recommend storing encryption keys through a separate air-gapped channel. This is not dissimilar to the "24-word wallet" for your cryptocurrency, which you would store safely, but in a business context.

Technology Choices and Nuances

As we have established the spectrum of topics for encryption and cryptography, I will be happy to share with you an initial list of technologies in **Table B**, where I would do further research. In the endnotes of this article, I will mention some nuances and cautionary words.

I am going to assume that we will use Windows as well as Linux environments, both on-premises and in the cloud.

Table B: Technology Choices

Category	Windows	Linux	Cloud Platforms
Policy & Cryptographic Controls	• Microsoft Security Compliance Toolkit[1] • Auditpol[2] • Microsoft Sentinel[3]	• OpenSCAP[4] • Auditd[5] • OSSEC[6]	• Cloud Security Posture Management (CSPM)[7] • Cloud Compliance Engines[8]
Encryption for Data at Rest	• BitLocker[9] • Encrypting File System[10] • Veeam Backup Encryption[11] • Microsoft Purview[12]	• dm-crypt[13] with LUKS[14] • Bacula[15] • VeraCrypt[16]	Similar to the encryption of data-at-rest for on-premises installations but with the additional complexity of key management (see below).
Encryption for Data in Transit	• Microsoft IIS (preferably with TLS 1.3 via GPO)[17] • OpenVPN[18] • WireGuard[19] • Microsoft Exchange TLS[20]	• OpenSSL (TLS 1.3 subset without AES128)[21] • OpenVPN[22] • WireGuard[23] • ~~Stunnel~~[24]	• SSL certificates offered by various cloud balancer products[25] • AWS Certificate Manager[26] • Azure Front Door[27]
Encryption for Data in Use	Please read the note on "Confidential Computing" above. • Windows Secure Enclave[28], AMD SEV[29], Intel SGX[30]		
Key Management (& HSM ?)	Please read the note on "Hardware Security Module" above. • Azure Key Vault with HSM[31], AWS CloudHSM[32], HashiCorp Vault[33], ~~GnuPG (GPG)~~[34], Thales HSM[35], nShield HSM[36] [37], and Fortanix.		
Certificate Management[2]	• Windows Certificate Authority • Certbot (Let's Encrypt) • Smallstep Cert Manager • Active Directory Certificate Services	• Certbot (Let's Encrypt) • Smallstep Cert Manager • CFSSL • Vault PKI Engine	• AWS Certificate Manager • Azure App Gateway • Google Certificate Authority Service

[2] Rob Stubb (thanks!) mentioned a few other names (Venafi, Keyfactor and AppViewX).

Category	Windows	Linux	Cloud Platforms
	• DigiCert Certificate Manager		
Cryptographic Updates & Monitoring[3]	• Windows Update • Microsoft Baseline Security Analyzer • Microsoft Defender for Endpoint • Qualys	• APT/YUM with Unattended Upgrades • Lynis • Wazuh • OpenVAS	• AWS Systems Manager • Azure Update Management • Google OS Config
Compliance & Audit Tools	• Microsoft Defender for Cloud • Azure Policy • Splunk • ArcSight	• ELK Stack • Wazuh • Nagios • Prometheus/Grafana	• AWS Security Hub • Azure Security Center • Google Security Command Center
Key Rotation & Lifecycle	• Microsoft Key Management Service • Azure Automation • PowerShell Scripts	• HashiCorp Vault Auto-Unseal • Automated GPG Key Rotation • Custom Shell Scripts	• AWS Secrets Manager • Azure Managed HSM • Google Secret Manager
Backup Encryption (List not exhaustive)	• Veeam • Veritas NetBackup • Azure Backup • Windows Server Backup (BitLocker)	• Amanda • Bacula • Duplicity	• AWS Backup • Azure Backup • Google Cloud Backup

[3] The field is expanding but specialised. Do your own research on cross-platform solutions (e.g. Infosec Global, Sandbox AQ, Fortanix). See my section on "Multi-OS Environment" under "Practical Considerations".

Practical Considerations:

Here are some additional points based on my personal experience.

Hybrid Environment:

Most of us work in a hybrid environment. While my lab is 100% cloud-native, all my remote management could be described as “on-premises.” Your business may also still have some on-premises servers. This means we all need to create a complete inventory of both cloud and on-premises assets and combine these if your tools don’t give you a unified view. I handle this manually, preferring to spend my limited budget on servers rather than extra tools. 😊

Multi-Jurisdiction Environment:

At the time of writing, my lab has servers on five continents (though not yet all eight 😊) covering six countries, some of which are outside the DORA jurisdiction. Mixing DORA and non-DORA regions isn’t an issue in itself, but we do need to be aware of cryptography restrictions in certain countries, such as China, Russia, India, the US (for export), and Brazil.

I’m not a lawyer, and this article won’t go into legal advice. But if I were to add infrastructure in one of these countries, I’d consult a specialist lawyer.

Multi-Cloud Environment:

Over the past five years, I’ve used over 20 cloud services and am currently using seven. In practice, running the same operating system (e.g., Debian) across different cloud providers isn’t always consistent. Using “containerisation” and “virtualisation” can help to standardise cryptography across systems but doesn’t entirely solve the issue.

My approach is to understand each cloud provider’s cryptography specifics and manage these separately from other technical challenges. For instance, one of my providers only supported RSA-2024 keys for SSH, whereas I prefer ed25519 and avoid RSA (even 4096-bit, but that’s another topic). Since they couldn’t update their settings, I ended our contract and switched providers. The cloud market is competitive, so there’s always an alternative that fits your needs.

Multi-OS Environment (including Legacy Technology Debt):

Legacy technology debt can be an impediment in DORA ICT 1774 compliance on encryption or for overall DORA, but that is not the only challenge.

I used to use too many operating systems, which was a nightmare to manage. I coined the term “Modern Technology Debt” to describe these self-inflicted issues. So, how did I solve it?

For now, I’ve strictly limited the number of OSs to keep things manageable: Arch Linux, Debian, and Ubuntu. I may add Fedora soon, as it will likely be the first OS to offer Quantum Resistant features. My remote access points are either Windows 11 Pro or Debian.

None of my servers run End of Life (EOL) or End of Support (EOS) systems, as I aim to upgrade at least six months before EOL.

Multi-Team Environment

I'm fortunate to work with only one other person who decides on cryptography and key generation—an external developer building my son's business infrastructure (which is also outside DORA's jurisdiction).

Because of this, controlling cryptography usage by two persons is easy, and I've created an air-gapped, immutable backup system to protect these keys. FYI – I use Veracrypt.

If you work with a large team or multiple third parties, you'll likely need better tools and a clear encryption policy. Be cautious using key management functions from major cloud providers and conduct your own risk assessment.

Native Encryption and Due Diligence:

I use BitLocker as native encryption for Windows 11 Pro endpoints, though, honestly, I don't fully trust it. Call it cyber intuition or paranoia, but I believe groups like APT28 (Fancy Bear) and APT29 (Cozy Bear) could potentially bypass it. Besides, who can be sure Android devices can't be rooted to override encryption?

Similarly, each of my cloud providers claims their server drives use native encryption. I hope they do, but as an individual customer, I have limited ability to conduct a full due diligence or negotiate terms.

As a business owner, your team might be able to negotiate better terms.

In my lab, I haven't yet found the perfect solution, but I'm currently researching ways to manage encryption keys used in transparent encryption remotely, instead of storing them on the same server. I'll be working on this over Christmas.

End Notes (and nuances)

[1] Download from Microsoft. https://www.microsoft.com/en-us/download/details.aspx?id=55319

[2] Learn at Microsoft Ignite. https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/auditpol

[3] Learn at Microsoft. https://learn.microsoft.com/en-us/azure/sentinel/overview?tabs=azure-portal

[4] Official source. https://www.open-scap.org/

[5] RedHat blog. https://www.redhat.com/en/blog/configure-linux-auditing-auditd

[6] Official source. https://www.ossec.net/

[7] Microsoft's explanation. https://www.microsoft.com/en-gb/security/business/security-101/what-is-cspm

[8] Vendor illustration (no registration required). https://cloudaware.com/technologies/compliance-engine/

[9] Learn at Microsoft Ignite. https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/

[10] TechTarget has a good explanation. https://www.techtarget.com/searchsecurity/definition/Encrypting-File-System

[11] Vendor illustration (no registration required). https://helpcenter.veeam.com/docs/backup/vsphere/encryption_backup_job.html?ver=120

[12] Learn at Microsoft Ignite. https://learn.microsoft.com/en-us/purview/information-protection

[13] See dm-crypt on GitLab wiki. https://gitlab.com/cryptsetup/cryptsetup/-/wikis/DMCrypt (Note: I prefer to use this for bootable USB keys for TAILS, laptops, desktops and on-premises servers)

[14] Wikipedia page for LUKS. https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup (Note: If using on cloud, you need to manage the key remotely.)

[15] Vendor explanation (no registration required). https://www.baculasystems.com/blog/backup-encryption-101/

[16] Official source. https://www.baculasystems.com/blog/backup-encryption-101/

[17] Microsoft Ignite provides guidance on configuration. https://learn.microsoft.com/en-us/windows-server/security/tls/manage-tls

[18] Community Edition exists for Windows. https://openvpn.net/community-downloads/

[19] Windows clients exist. https://www.wireguard.com/install/

[20] An older explanation is available here. https://techcommunity.microsoft.com/blog/exchange/exchange-server-tls-guidance-part-2-enabling-tls-1-2-and-identifying-clients-not/607761 However, I am aware that Microsoft has made further improvements that you can search.

[21] Please note that I prefer to achieve the cipher selection via Nginx. See my suggestions on the GitHub thread here. https://gist.github.com/gavinhungry/7a67174c18085f4a23eb (my handle is BeatQuantum).

[22] The Ubuntu installation guide is good enough. https://ubuntu.com/server/docs/how-to-install-and-use-openvpn (If you get stuck, ask Copilot or check LinuxBabe's tutorial)

[23] The official source. https://www.wireguard.com/install/ (Please note some Linux repository packages may have an older version).

[24] I have included stunnel for completeness, however I do not use it. Instead, I prefer to use WireGuard for peer-to-peer encryption and TLS 1.3 on my webserver (HTTPS) and DoT server (850/tcp).

[25] Please note I prefer SSL/TLS passthrough to preserve end-to-end encryption. This means the load balancer only does what it says on the tin but that is my personal preference. If you require packet inspection and WAF at the perimeter, you can use edge certificates from various providers or Cloudflare.

[26] Vendor description (requires no registration). https://aws.amazon.com/certificate-manager/

[27] Microsoft explanation. https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview (It combines multiple functions).

[28] Microsoft Ignite explains how to secure SQL traffic. https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-enclaves?view=sql-server-ver16 Caution: If using for the OS, please check the compatibility with your antimalware/antivirus solution.

[29] AMD explanation is available here. https://www.amd.com/en/developer/sev.html Caution: You will need expertise to enable this on Windows. See Microsoft Ignite guide here. https://learn.microsoft.com/en-us/troubleshoot/windows-server/setup-upgrade-and-drivers/windows-server-support-installation-for-amd-role-family-processor

[30] Microsoft Ignite article on Intel SGX. https://learn.microsoft.com/en-us/azure/confidential-computing/virtual-machine-solutions-sgx

[31] Microsoft explanation is here. https://learn.microsoft.com/en-us/azure/key-vault/managed-hsm/mhsm-control-data .

[32] Vendor explanation (requires no registration). https://aws.amazon.com/cloudhsm/

[33] Note that Hashicorp can be used in conjunction with other HSM based solutions.

[34] Caution: GPG can be a strong tool for data encryption and digital signatures in environments that don't demand centralised key management or hardware-based protection. However, for enterprise-grade encryption or compliance-driven environments, where centralised control, auditing, and possibly hardware-backed security are required, GPG alone might fall short.

[35] Vendor explanation (requires no registration). https://cpl.thalesgroup.com/en-gb/encryption/hardware-security-modules

[36] Vendor explanation (requires no registration). https://www.entrust.com/products/hsm

[37] See Linux OS compatibility at the Vendor page (requires no registration). https://nshielddocs.entrust.com/app-notes/pcie-hsm-compatibility/intro.html