

Dispelling Mainframe Security Myths:

# A Guide to Better Data Protection

**JOE STURONAS**

CHIEF TECHNOLOGY OFFICER, PKWARE

## Contents

<b>DISPELLING MAINFRAME SECURITY MYTHS: A GUIDE TO BETTER DATA PROTECTION</b> .....	3
<b>MAINFRAME SECURITY MYTHS</b> .....	4
MYTH 1: Using hardware crypto on the mainframe prevents data from being decrypted on other platforms.....	4
MYTH 2: We did not purchase crypto cards, so we can't perform hardware crypto on the mainframe. ....	5
MYTH 3: ICSF performs hardware crypto, so I can just use ICSF to encrypt files without additional software. ....	5
MYTH 4: I don't need file encryption because my DS8000 performs disk encryption, so my files are safe. ....	6
<b>REDUCING THE OVERHEAD OF STRONG CRYPTOGRAPHY</b> .....	6
<b>PROTECTING &amp; MANAGING CRYPTOGRAPHIC KEYS</b> .....	7
<b>KEY MODE PROCESSING</b> .....	9
Clear Key .....	9
Secure Key .....	9
Protected Key .....	9
<b>ENFORCING DATA PROTECTION POLICY</b> .....	10
<b>CONTINGENCY KEY</b> .....	10
<b>ENCRYPTION AS A STANDARD: FIPS 140-2</b> .....	11
<b>A BETTER WAY</b> .....	12

## Dispelling Mainframe Security Myths: A Guide to Better Data Protection

In Medieval times, people lived in large stone castles and walled cities to protect themselves from intruders. Protection of citizens and royals inside of these walls focused on strong perimeters: walls were tall and difficult to scale, a drawbridge closed to unwanted outsiders, moats surrounded the walls and hot oil was even poured on those who got too close.

In response, intruders developed new tactics of attack and used the latest technology to blast away at the presumably strong perimeter walls.

Fast-forward about 1,000 years and it might seem that the scenarios for protecting valuable enterprise data are entrenched in the strategies of the past. Today, organizations are spending their limited security budget on strengthening enterprise walls and fortifying access—without realizing that the changing nature of attacks puts their data and business at risk well beyond the traditional perimeters.

Focusing only on perimeter security is a battle better suited for bygone times. Nowadays, sensitive data regularly moves from platform to platform and from endpoint to endpoint, inside and outside the organization. Some take the approach of securing endpoints and network connections. However, the reality is that these strategies leave data exposed at certain points in the storage and transfer process. In a data-centric security approach the data itself is protected and is not dependent upon the individual endpoint or network security scheme. Data-centric security involves protecting the data itself through the use of data encryption and authentication that are enforced by policy administration.

Organizations recognize the need to protect the data itself in order to ensure its security as it moves beyond the perimeter; however they struggle with the additional cost to their operations. For some, the combination of the operational overhead of encryption, the need for secure encryption key generation and management, and the burden of appropriate data security policy enforcement introduce new challenges. Fortunately, there are ways these costs can be contained, and in some cases, even reduced to a have negligible impact on operational budget.

IBM®'s z/OS® and the mainframe hardware supporting it form the foundation for efficient, strong data encryption, safe key generation and management, and sound data protection policy enforcement. Combined, these capabilities offer organizations a cost-effective option for safeguarding digital assets. The challenge lies in protecting sensitive data as it moves off the mainframe and outside the perimeter of the organization.

SecureZIP complements z/OS mainframe crypto capabilities by encrypting sensitive data at the file level so that it is secure while at rest and in transit, as it leaves the organization and moves across computing platforms and environments.

In this guide, we will dispel some of the common myths associated with System z cryptographic facilities and explore encryption strategies that ensure data is protected not only on the mainframe but as it moves to other computing platforms and environments.

## Mainframe Security Myths







### MYTH 1

#### Using hardware crypto on the mainframe prevents data from being decrypted on other platforms.

There are System z® facilities, such as hardware crypto, which are not special purpose engines but provide dramatic performance benefits and reduction in general CP utilization. Many people confuse hardware crypto for a special purpose engine, but it is not. CPACF stands for “Central Processor Assist for Cryptographic Function,” and it does just that. It accelerates crypto operations so they use less general CP and greatly increase elapsed time. CPACF does not require an additional investment and comes with every System z mainframe. It can be activated with the free Web download ICSF (Integrated Cryptographic Service Facility). Products like SecureZIP for z/OS take advantage of hardware crypto through CPACF and the ICSF interface to secure data as it moves off the mainframe to other computing platforms. Because the ZIP and OpenPGP standards operate independently of platform and are portable across all major computing systems, encrypted ZIP and OpenPGP archives created on System z can be transferred to System i, UNIX (HP-UX, Solaris, AIX), Linux, Windows® Server, Windows Desktop or Mobile (Android® and iOS®) and be decrypted on any of those platforms, or vice versa.



Encrypted ZIP and OpenPGP archives created on System z can be transferred to System i, UNIX (HP-UX, Solaris, AIX), Linux, Windows Server, Windows Desktop or Mobile (Android and iOS) and be decrypted on any of those platforms, or vice versa.

						
	<b>z10-EC 2097</b>	<b>z10-BC 2098</b>	<b>z196 2817</b>	<b>z114 2818</b>	<b>zEC12 2827</b>	<b>zBC12 2828</b>
<b>Supported Algorithms</b>	DES, 3DES, AES128, AES192, AES256					
<b>Crypto Hardware</b>	CPACF CEX2C CEX3C	CPACF CEX2C CEX3C	CPACF CEX3C	CPACF CEX3C	CPACF CEX3C CEX4C	CPACF CEX3C CEX4C

**MYTH 2**  
**We did not purchase crypto cards, so we can't perform hardware crypto on the mainframe.**

Many people believe that in order to perform hardware cryptographic operations such as encryption and decryption, they first need to purchase hardware crypto cards like the Crypto Express 3 (CEX3) or Crypto Express 4S (CEX4S). While it does depend on the type of key mode you want to process in, you can perform clear key processing with CPACF, which is hardware crypto on the CP itself that comes with every modern System z CEC today (starting with z9). When sensitive data needs to move off the mainframe, SecureZIP for z/OS takes advantage of CPACF in clear key mode, as well as Crypto Express 3 and Crypto Express 4S in secure key mode. SecureZIP can even take advantage of protected key mode utilizing CPACF and Crypto Express 3/Crypto Express 4S. We will take a closer look at key mode processing available on z/OS a little later.

The figure above shows the hardware crypto capabilities that exist on each of the modern mainframes, as well as the crypto algorithms supported by SecureZIP for z/OS.

**MYTH 3**  
**ICSF performs hardware crypto, so I can just use ICSF to encrypt files without additional software.**

ICSF works with the hardware cryptographic features and the Security Server (RACF, ACF2 and Top Secret) to provide high-speed cryptographic services in the z/OS environment. ICSF provides the API (Application Programming Interfaces) by which applications request the cryptographic services.

ICSF can be used with SecureZIP for z/OS, to provide file-based encryption, but ICSF cannot encrypt a file by itself. ICSF provides the basic fundamental building blocks to access the hardware crypto facilities, but lacks the application elements such as file handling, file processing and user interface. It would be analogous to buying a car engine and expecting to just drive the engine. The chassis, tires, seats, dashboard and body represent the application elements, while ICSF is the engine that drives the cryptographic functions.

**MYTH 4****I don't need file encryption because my DS8000 performs disk encryption, so my files are safe.**

The only type of encryption the DS8000 (and similar mainframe storage subsystems) provides is AES-256 bit disk encryption. Disk encryption means that whole disk encryption is performed on the disks that comprise the storage subsystem. It protects the disk, so if the disk needs to be replaced, the disk can be removed, but all the data on the disk cannot be accessed unless the private key to decrypt the data is known. Then, as files are read off the storage subsystem, they are automatically decrypted and left unprotected. Conversely, when files are being written to the storage subsystem, they are automatically encrypted.

Storage subsystem disk encryption is another type of encryption and is analogous to whole disk encryption on distributed systems. It provides a semblance of encryption at the device level. Whole disk encryption (as well as folder encryption, such as that provided by Microsoft® BitLocker®) is useful when a laptop is powered down and falls into the wrong hands. However, when the laptop is powered on and the operating system is up and running, whole disk and folder encryption provide no protection. Files are automatically decrypted as they are accessed and moved off the laptop. The same is true for storage subsystem disk encryption.

While disk encryption protects the disk, it leaves sensitive data vulnerable when it moves off the disk. File encryption is a more granular way to encrypt files than whole disk encryption. File encryption focuses on the file itself, meaning that the encryption's security stays with the file, even if it is being read from the storage subsystem.

## Reducing the Overhead of Strong Cryptography

Encryption, by its nature, is computationally intensive. The work required to randomize data in a way that allows it to be later restored to a readable state far exceeds the effort required to simply write the data from one location to another. This is commonly an obstacle that prevents many organizations from including encryption in their data workflows. The mainframe addresses this need with hardware cryptographic acceleration.

Unlike many other contemporary computing platforms, IBM zEC12 and zBC12 mainframes include

hardware-based cryptographic acceleration natively and offer options to reduce the capacity required to encipher data. Hardware-based cryptography (encryption/decryption, hashing, and PRNG performed using direct calls to the hardware's instruction set) always requires fewer resources than cryptography performed using software alone. It avoids the overhead of system management, command interpretation and other operations required when processing software applications. Contemporary IBM mainframes offer at least three approaches for reducing the cryptographic calculation load for a given operation:

CPACF—Central Processor Assist for Cryptographic Function

CEX3C—Crypto Express 3 Coprocessor

CEX4C—Crypto Express 4 Coprocessor

In all cases, the hardware processes the largest majority of the cryptographic calculations, reducing the burden on the general purpose CP by an order of magnitude compared to the same operation performed in software.

For example, assume that a 2098-P04 mainframe encrypts a one-gigabyte transaction log for secure storage five times each hour. Using the IBM software encryption available through the ICSF, 100.91 CPU seconds are required. Using the same machine and the IBM CPACF hardware encryption acceleration available directly from the CP instruction set, the same job requires only 14.90 CPU seconds, a difference of 86.01 CPU seconds per run. Assuming this process continues during a given year, use of the hardware encryption facility could drastically reduce the amount of capacity that is required for a given machine or LPAR.

## Protecting & Managing Cryptographic Keys

Encryption efficiency is immaterial without the availability of durable encryption keys and the diligent protection of those keys. Devising the appropriate scheme to protect keys and to ensure that key use is controlled in an auditable way has defeated many early implementations. Both the cost of developing the necessary scheme ad hoc and the cost of purchasing packaged software products for this purpose have been seen as burdens too great to bear. Moreover, most (if not all) packaged solutions to this point either failed to cover all types of keys or sat outside the rigorous protection available on the mainframe.

Practically speaking, keys for both encryption and decryption are required for every encryption use, and such paired keys come in two different types: symmetric and asymmetric. With symmetric keys, the key used for encryption and decryption remains the same. Think

of it as a password or a shared secret between the encrypting person or organization and the decrypting entity. This presents an obvious risk of insider compromise—if an encrypting operator knows the password or passphrase, it can be used maliciously. The operator could, for instance, sell the passphrase to a third party to make a copy of the encrypted data and decrypt it for fraudulent use later and elsewhere. Maintaining a separation of duties between a security professional who manages passphrases and the operators who execute jobs that require them can be a significant challenge, which is multiplied when the additional requirements of appropriate logging for audit and compliance review are added. Building such functionality is onerous, particularly because this area of information technology is outside of all but a few organizations' core competencies. Building that competency in-house or contracting for it can be very expensive.

However, separating passphrase management from job execution is possible with the IBM mainframe. Machines that include the Crypto Express 3 or 4S cards with ICSF offer the capability to segregate key management from key use via the Cryptographic Key Data Set (CKDS). In the real applications of business data, the facilities that support this capability generally lie outside the areas most system programmers have worked with. Learning to use the interfaces correctly and in a manner that satisfies internal and external auditors can be very time consuming and prone to error. The second type of encryption/decryption keys, asymmetric keys, addresses many of those learning curve and cost issues. Public key cryptography, relies on a branch of mathematics focused on factoring prime numbers, and can relate two keys in a manner whereby knowledge of one key in no way provides any means for deriving or reverse engineering the companion key of the pair. This greatly mitigates the risks of key management. An operator can have knowledge of and access to a public key used to encrypt sensitive data for a recipient but has no access to the private key needed for decryption. Therefore, the operator has nothing of value to use or sell to a malicious third party.

However, the applications that implement the complex prime number mathematics, called certificate authorities, have been expensive to license and/or difficult to secure since they were implemented for open system architectures. The latter is a crucial issue—if a certificate authority on a Windows® or UNIX platform is compromised, a data thief could have access to all the key pairs it has generated. Again, concerned professionals seek a more cost effective approach that also offers the appropriate level of security. IBM recognized this and provides the foundation to

address it. A certificate authority is included in the operating system at no additional cost and brings the benefit of “gold standard” protection and logging for audit of the mainframe security servers. The key generation capabilities with their associated centralized key stores can be used by a variety of mainframe applications, greatly reducing the overhead of administering proprietary key stores.

There are cases for both symmetric and asymmetric keys, even though asymmetric keys are typically more durable. The most common use case for symmetric keys is when data leaves the organization and it needs to be protected data at rest and in motion. There are means to send data through security transport tunnels, such as TLS (Transport Layer Security) or SSL (Secure Socket Layer). While TLS protects the data while in motion in the tunnel, it does not protect it at rest, which is now a more common requirement. The data is exposed before it is sent, and exposed when it lands at its destination. In this use case, the encryption is temporary and a symmetric key is used to encrypt the data. Then, the recipient of the data would use the same symmetric key to decrypt the data and then re-encrypt the data with their own key management, audits and policy mechanisms. The encrypted file is very temporary and not required for long term archive or access. This is a perfect use case for symmetric keys.



## Key Mode Processing

Here are the three primary key mode processing models:

### CLEAR KEY

CPACF is hardware crypto that supports clear key processing, which is a somewhat derogatory name, even if it proves to be descriptive. With CPACF, there is a small period of time where the private key is exposed in the address space, and if someone were to dump the address space, the private key could be exposed. This is why it is called clear key.

### SECURE KEY

In contrast, the Crypto Express 3 and Crypto Express 4S facilitate secure key processing, which means that the private key is only exposed in the tamper resistant protection of the Crypto Express 3 or 4S, and not in the address space. If the address space using the Crypto Express 3 or 4S were ever to be released, the private key would never be exposed, because it would not release the contents of the Crypto Express 3 or 4S. Like CPACF, the Crypto Express 3 or 4S use very little general CP, but can suffer in elapsed time performance degradation due to the latency of the access to the card through the PCIe bus, which is slower than access through the chip found with CPACF.

### PROTECTED KEY

There is another model of crypto processing that is a hybrid of clear key and secure key, called protected key. Protected key represents the best of both worlds, because it has the elapsed time speed of CPACF and the private key security of protected key. Protected key processing requires CPACF and a Crypto Express 3 or 4S. It takes advantage of a change that was made in the z10 model that increased the size of the HSA (Hardware Storage Area) and allows for atomic operations to be executed on the private key material stored in the HSA. Because that area of the HSA is highly protected, neither an address space dump nor stand alone dump would expose the private key. It is not tamper resistant like the Crypto Express 4S, but the private key is decrypted in the Crypto Express 4S and through special instructions, moved to the protected area of the HSA.

What is interesting about protected key processing is that the elapsed time benchmarks for protected key are actually faster than clear key processing, most likely due to IBM's optimization in the instructions for protected key. SecureZIP for z/OS supports clear key, secure key and protected key processing so that crypto operations have a negligible general CP overhead, while providing flexibility on the key mode processing which may be required for compliance and regulatory mandates.

SecureZIP for z/OS is the only file encryption application on z/OS that supports all three key mode processing models. Most file encryption software applications on z/OS only support clear key operations, because if they support hardware crypto, they only support CPACF.

## Enforcing Data Protection Policy

Encryption is a powerful tool and, like all powerful tools, it can cause great damage if not used correctly. Used appropriately, it addresses specific needs, mitigating risks to the confidentiality of sensitive information. Used inappropriately, data could be encrypted with a passphrase and held for ransom, something that has been documented as an occurrence by outside attackers with ransomware such as CryptoLocker. It could even happen with a disgruntled insider who accesses to encryption software licensed by the employer. Encryption must be subject to appropriate control and supervision to be useful to the organization.

Moreover, control alone is not enough. Not only must the organization ensure that appropriate oversight is imposed, it must also guarantee that such oversight includes appropriate logging of actions so that they

can be audited at a later time. Each change to security policy must be documented, including the time, date, operation applied and who initiated the change. If a compliance officer or an auditor requests proof that the appropriate controls are in place and are having the desired effect, it is imperative that the organization have the appropriate records at hand to ensure that data security policies are in place and that they cannot be changed or circumvented.

While the security servers available to z/OS provide the infrastructure for satisfying these needs, imposing the control on the native IBM encryption facilities—or to packaged applications that are not specifically integrated with them—poses an arduous, expensive and complex effort.

---

## Contingency Key

A contingency key is a private key held by the organization using their current key management/access methodology, whereby designated individuals have the corresponding private keys that allow for decryption of data for contingency access. The contingency key holders could be the InfoSec team from an organization-wide perspective, or departmental owners at a more granular level.

Policy settings do not have to be the same throughout the mainframe environment. Policy management can be refined to the point where each business unit or department has their own policy settings, and thus their own contingency keys. Policy management enforces

that contingency keys are used effectively throughout the organization.

Protecting data through encryption without policy management and contingency keys can be dangerous and reckless. Encryption without proper policy management puts the organization at risk of quickly losing control of their data. Employees will be able to encrypt, but if someone leaves or is a “bad actor,” the organization may lose access to that encrypted data. Focusing on data protection really means implementing controlled encryption, with policies in place that include contingency keys with every encryption operation so that the organization never loses control of the data,

even if someone leaves. This policy is enforced by the security server in use (RACF, ACF2 or Top Secret) and audited by SMF.

Strong encryption with no policy-based contingency key creates a higher risk of lost data. For instance: A financial services company allowed mainframe applications to encrypt data with a product incapable of providing policy-based contingency keys or private key escrow. By allowing the use of this product, they were left with gigabytes of encrypted, inaccessible, useless data, because they did not have the passphrases needed to encrypt the data and the people with that information had left the organization. If they had only implemented a policy based encryption solution, all of that data would have been protected and accessible.

Security minded companies should be cautious when evaluating encryption solutions. Solutions like MegaCryption, SDS E-Business Server, Data21 and Encryption Facility for z/OS enable organizations to secure data with password-based encryption, but they do not have policy management or contingency key capabilities. An enterprise security product like SecureZIP for z/OS provides the ability to centrally manage encryption through policy, so that one or more contingency keys can be applied to every encryption operation throughout the enterprise. This gives the organization peace of mind in how encryption is being used and the ability to access encrypted data for audit or recovery purposes.

---

## Encryption as a Standard: FIPS 140-2

Most U.S. government agencies and the private sector companies that work with them (like banks and healthcare insurers) are required to encrypt data with software that meets Federal Information Processing Standards (FIPS) 140-2 compliance.

Data encryption software is FIPS 140-2 compliant when it uses cryptographic algorithms that have been validated through the U.S. National Institute of Standards and Technology's (NIST) Cryptographic Module Validation Program (CMVP). Cryptographic modules that have been validated are issued a certificate number. Only software that is able to identify the NIST certificate number for the cryptographic algorithm's FIPS 140-2 validation can qualify as being FIPS 140-2 compliant.

SecureZIP for z/OS is FIPS 140-2 compliant and it lists corresponding FIPS 140-2 validated cryptographic algorithm certificate numbers. Not all FIPS levels are the same. For instance, SDS E-Business Server is FIPS 197 compliant, which does not provide the same crypto depth and standard as FIPS 140-2. FIPS 197 addresses just the AES algorithm and does not address other, more comprehensive requirements found with FIPS 140-2. ASPG's MegaCryption claims to be "FIPS Compliant," though does not reference any specific level achieved through the compliance standard, which makes it ambiguous. It doesn't have a FIPS mode option and never references compliance at the FIPS 140-2 level. For data encryption software to be FIPS 140-2 compliant, while in FIPS mode it must list the

FIPS 140-2 validated cryptographic libraries that it uses along with the certificate numbers.

SecureZIP for z/OS is FIPS 140-2 compliant on the zEC12 with a Crypto Express 4S card configured as a coprocessor, and with SecureZIP for z/OS configured in FIPS 140-2 mode, which is associated with NIST FIPS 140 Certificate #1505. If the encryption software does not support FIPS 140-2 mode, there is no way to impose policy and ensure the encryption operations are FIPS 140-2 compliant.

A payments processing company that handles a lot of sensitive information for its client, the U.S. federal government, chooses to encrypt only in FIPS 140-2 mode, making all their encryption FIPS 140-2 compliant. When in FIPS mode, they run in Secure Key Mode and use only AES 256-bit encryption and, because of that, they do all the encryption work on the Crypto Express 4S card.

---

## A Better Way

PKWARE has provided applications tailored for the mainframe data center for more than 20 years, natively implementing first for MVS, then OS/390, and now for z/OS. The version 15 release of SecureZIP focuses on meeting the market needs described above. It provides a cost-effective, easily integrated packaged solution that enables organizations to integrate encryption and key management facilities into existing and new workflows. In addition, it offers fully supported integration with the IBM facilities for encryption acceleration, key generation and management. Here are three benefits from encryption policy configuration, enforcement and oversight:

- ▶ Reduce administrative burden—By using the common key repository used by other applications, SecureZIP for z/OS v15 can reduce key management effort, thereby reducing administration and expense.
- ▶ Reduce risk—Separation of duties for passphrase management provides segregation of roles between the security administrator and the systems engineer.
- ▶ Protecte data—Superior policy control and policy change audit elevate data security.

SecureZIP for z/OS v15 enables an organization to standardize on a single set application that integrates naturally into existing and new workflows. Using this packaged product, organizations can quickly protect sensitive data that may be stored locally, on media, or exchanged across operating system boundaries, across geographically dispersed locations, or with customers, vendors and partners.

It is a somewhat confused fact, that you can't compress encrypted data. If data is encrypted, by definition it is highly randomized, and randomized data can't be compressed. If you are going to encrypt data, then you might as well compress the data at the same time, in the same pass-through, because you won't get another opportunity to do so.

In the spirit of reducing general CP overhead, SecureZIP for z/OS (version 14 and higher) now supports the zIIP processor for archive management. SecureZIP for z/OS has made compression and CRC (Cyclical Redundancy Check) zIIP eligible. This means that when SecureZIP for z/OS compresses and encrypts a file, approximately 90% of the CPU workload is zIIP eligible and only 10% is processed by the general CPs.

SecureZIP for z/OS v15 is able to reduce the total general CP to 5 seconds, down from 31 seconds when compressing and encrypt-

ing a 2GB file using a zIIP special purpose engine by making the deflate compression algorithm and CRC (Cyclical Redundancy Check) zIIP eligible. That results in an 81% reduction in chargeable CPU over SecureZIP for z/OS software based solution and a 92% improvement in CPU time over the leading competitor.

IBM introduced a new PCIe card in late 2013 called the zEnterprise Data Compression (zEDC). This card has one purpose which is to perform deflate compression.

It does this very quickly and scales nicely. Like the zIIP special purpose engine, the zEDC is not considered as chargeable capacity like a general CP, but can provide a very specialized workload capability for compression. The zEDC is very similar to the Crypto Express cards (CEX3C, CEX4C), which are also PCIe cards that contain their own processor and perform specialized cryptographic work in the form of off-board processing from the CPs.

Best practices and expected standards of care for data protection continue to evolve, as both the risks are more clearly identified and the monetary impact of data breaches are more accurately estimated. All organizations seek a better way—a better way to protect sensitive data with strong encryption while still remaining operationally efficient, a better way to protect and manage the keys necessary to ensure that the encryption applied is sufficiently durable to resist attack, and a better way to impose appropriate data encryption policy reliably. For your mainframe data, SecureZIP for z/OS is that better way.