PKWARE®

Data Security Guide:

Protecting Sensitive Data in an Open Systems Environment

MICHAEL LAZAR SENIOR SALES ENGINEER, PKWARE

PKWARE®

Data Security Guide:

Protecting Sensitive Data in an Open Systems Environment

MICHAEL LAZAR SENIOR SALES ENGINEER, PKWARE

Contents

Data Security Guide:

Protecting Sensitive Data in an Open Systems Environment

Servers provide the basis for sharing and moving information throughout the enterprise, as well as with external business partners. Every day, a staggering amount of data flows through the data center, much of it is sensitive in nature. When data processing happens on the mainframe, in many cases, it is then sent to distributed platforms for more cost-effective processing before being sent to customer and partner destinations. While this flow of data between platforms creates efficiency, it also exposes security vulnerabilities that have the potential of leading to failed compliance or worse yet, a breach.

In order to address compliance requirements and avoid the risk associated with a breach, security minded professionals turn to encryption. There are however, many types of encryption. Some secure the device, while others protect the pipe. The choices are daunting and it is imperative to consider the life cycle of data—from origin to endpoint. One missed step and your organization's data could be exposed and wind up the next breach news headline. How can you be sure the security you have is good enough?

It is important for those tasked with maintaining data security in the organization to understand how data moves within and outside the enterprise, learn who has access to sensitive data, and figure out how to control it, all without sacrificing efficiency.

Data-centric encryption protects information on UNIX, Linux, and Windows® servers without disrupting existing operations and processes. In this guide, we will explore how to protect sensitive data in an open systems environment in order to meet compliance requirements and protect against a breach. We will also provide helpful tips to streamline deployment and administer policy, along these five critical considerations:

1. Encryption theory
2. Enterprise deployment using configuration files
3. Application integration
4. Addressing uncontrolled encryption
5. Going beyond compliance

Encryption Theory

Data encryption can be categorized into two groups: symmetric key encryption and asymmetric key encryption (also known as public key encryption).

SYMMETRIC KEY ENCRYPTION

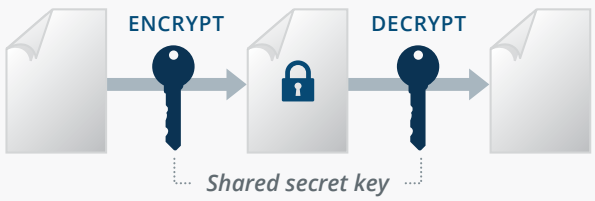


FIGURE 1—THE FLOW OF A MESSAGE PROTECTED USING SYMMETRIC KEY ENCRYPTION.

ASYMMETRIC KEY ENCRYPTION

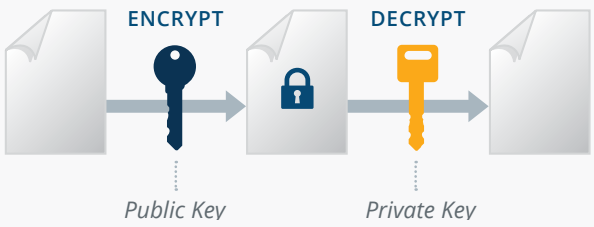


FIGURE 2—THE FLOW OF A MESSAGE PROTECTED USING ASYMMETRIC, OR PUBLIC KEY ENCRYPTION.

Symmetric key encryption is most commonly associated with password or passphrase based encryption. Figure 1, above, shows how the same key is used for encryption and decryption. Symmetric key encryption works best for non-persistent data, or static, non-transactional data.

Non-persistent data is typically encrypted with a symmetric key (passphrase) and sent to another entity for use. In this way the data is protected at rest before it is sent, while it is in motion and when it reaches the data consumer. There is no need for the data to persist.

However, symmetric based encryption does not scale well, particularly when the data needs to persist or it needs to be shared with multiple recipients. Another issue with symmetric key encryption is that companies often end up dealing with so many different passphrases that they are left with uncontrolled encryption. The origins of these uncontrolled passphrases range from employees that have left the organization, to partner exchanges that have gone stale, to rogue employee behavior. We'll dive into uncontrolled encryption later on.

Alternately, when data needs to be protected for longer periods of time for compliance or regulatory purposes and when it is going to be shared with multiple sets of recipients then the best option is asymmetric keys, otherwise known as public key encryption (Figure 2). Public key encryption uses both a public key and a private key. The public key is used for encryption and authentication while the private key is used for decryption and digital signing. The two keys are mathematically related through the use of cool math including prime integer factorization, discrete logarithm and elliptic curve relationships. The strength of the encryption is based on the computational intensity that it would take to determine the private key. The public key should be easily accessible to any authorized user, and the private key should be kept private and protected.

One or more public keys can be used to encrypt data and any of the corresponding private keys are able to decrypt the data. (Public key encryption is also a critical component of contingency key and policy administration, which we address later on).

PROTECTING DATA ON THE MOVE

The nature of business today requires data to move across a complex network of devices and computing platforms as well as physical, virtual and cloud environments. Here is a brief overview of device and transport security strategies, including ideas to keep your data from being exposed.

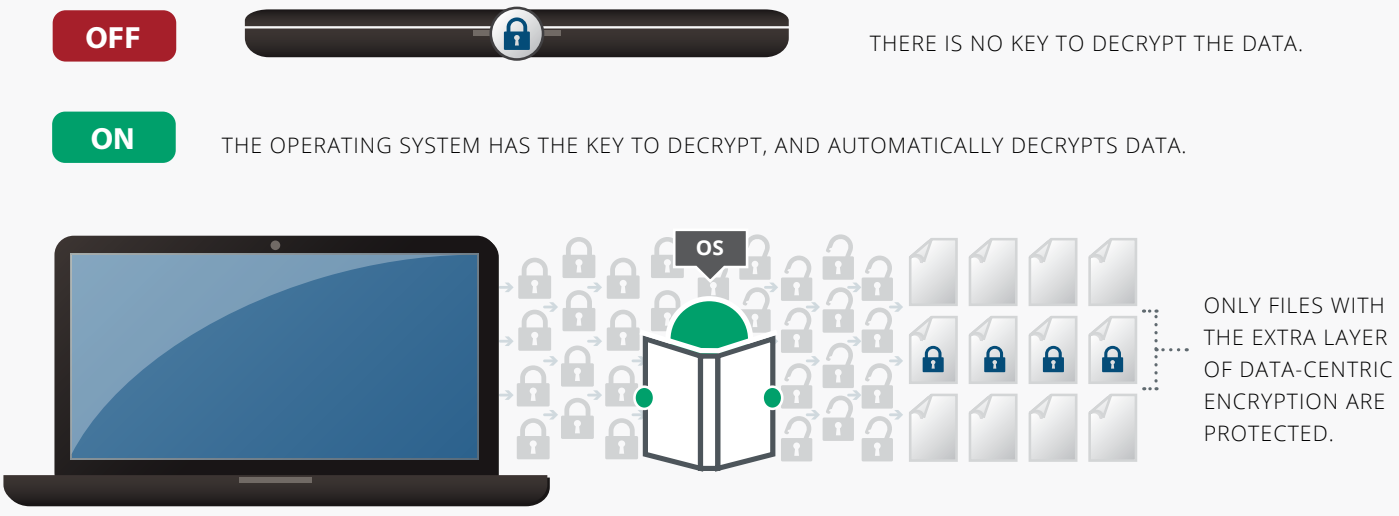
Whole Disk Encryption (WDE) provides a semblance of encryption at the device level and is useful when a powered down endpoint falls into the wrong hands. An "endpoint" is any extended enterprise network device where there is physical storage and the network device contains enterprise data (examples include server, desktop, laptop, tablet, mobile phone). Because the data cannot be accessed without the proper credentials, it is presumed that the data is "safe".

However, when the endpoint is powered on and the operating system is up and running, WDE, as well as similar methods like folder encryption, provide no protection. Files are automatically decrypted as they are accessed and moved off the endpoint (Figure 3). The data in motion is unencrypted—and thus subject to the same risks as if it were not encrypted at all.

Transport Layer Security (TLS) is a protocol that provides privacy between communicating applications and their users on the Internet. With TLS, no third party may eavesdrop or tamper with any message. However, protecting data-in-transit via the 128-bit SSL encryption of TLS only provides security during transit not at its origin or destination.

Data-centric security is the best way to protect sensitive information because it employs file-level encryption that is portable across all computing platforms, operating systems and cloud computing environments. Data is protected regardless of where it resides, where it goes, how it is stored or the transfer protocol.

FIGURE 3—WHOLE DISK ENCRYPTION SECURITY WHEN OPERATING SYSTEM IS POWERED ON VS. OFF.



In the case of WDE, data-centric security protects the data when the endpoint is powered off as well as when the operating system is running. Data in an encrypted ZIP or OpenPGP file remains protected as it is copied off the endpoint. WDE and folder encryption are useful as another layer in the "defense in depth" security approach, which aims to protect data at rest, data in motion and data in use. Data-centric protection through encryption renders the data unusable to anyone who does not have the key to decrypt it. The owner of the decryption keys maintains complete control over the security of that data and determines access to that data. When WDE, folder or TLS encryption are used on their own, they don't provide true data-centric security.

Enterprise Deployment Using Configuration Files

Many organizations view deploying enterprise-wide, data-centric security as a daunting task. The reality is that with capabilities like alternate configuration file integration, data-centric security has become quite manageable when it comes to enterprise rollout and ongoing maintenance. PKWARE's SecureZIP for Server provides data-centric security with an alternate configuration file option that allows centralized control of default settings and default inputs for scripts and applications. Organizations can avoid the common issue of "hard-coding" passwords into scripts, because only the intended scripts can read a configuration file that contains sensitive information.

By separating the jobs of securing data and configuring the security controls, organizations can enforce best practices for securing sensitive data. Centrally administering the scripts and application integration points allows multiple departments to roll out security that is aligned with enterprise best practices. Only the intended scripts can read a configuration file that contains sensitive information because configuration files may contain passphrases used in encryption or decryption operations (see Figure 4, right). These configuration files may be protected separately from the scripts that contain the SecureZIP commands.

From a maintenance perspective, once deployed, the alternate configuration files enable an organization to make centralized changes in an immediate and uniform way.

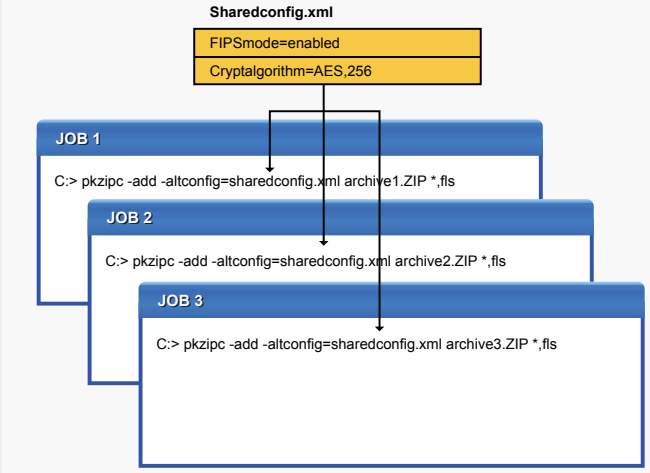


FIGURE 4—A SINGLE CONFIGURATION CHANGE APPLIED TO MULTIPLE JOBS.

PROTECTING ALTERNATE CONFIGURATION FILES

It is important to maintain control over configuration files. Allowing unauthorized users to change the alternate configuration file creates the same risk as using a default configuration that could be unintentionally modified. Protecting the directory containing the alternate configuration files is done easily using Access Control Lists. Simply create a ZipConfig group that owns all alternate configuration files and provide only read access to unauthorized users.

Application Integration

Many job processes in the server environment require data to be decrypted and staged, unprotected, in order for the application to access it when needed. Once data processing is complete, the application again stages the data to disk, leaving it unprotected. This creates significant security risk.

With SecureZIP for Server's Application Integration, encrypted data is streamed directly in and out of applications, eliminating the need to stage it to disk. This keeps the data secure during processing as well as improves operational efficiency by reducing the number of steps needed to process data.

SecureZIP provides robust integration for new and existing applications and multiple methods for integration. By using the operating system's inter-process communication facilities, named pipes, unnamed pipes and standard stream-oriented I/O channels can securely move data from one to another. With pipes you can stream ZIP files or file contents directly to and from SecureZIP, providing flexibility in constructing processing pipelines to meet application requirements for streaming data. For Linux Systems, the scripts below show how to create a simple named pipe to connect SecureZIP with other processes.

```
mkfifo ReportPipe
SQLQuery —outputfile=ReportPipe & pkzipc —add —password="pkwarepkware" -recipient=PKWAREPartner-
LinkCA@pkware.com -cryptalgorithm=AES,256 —filetype=pipe —stream Report.ZIP ReportPipe
```

On Windows systems, the script below illustrates the simple use of streams in Visual Basic. The script reads a stream from STDIN and writes the data to a file. The data stream comes from SecureZIP, which decrypts and pipes the data to the script from an encrypted ZIP archive.

```
Dim streamIn
Set streamIn = WScript.StdIn
Const ForWriting = 2 ‘ Input OutPut mode Const Create = True
Dim MyFile
Dim FSO ‘ FileSystemObject
Dim TSO ‘ TextStreamObject
MyFile = “c:\download\test\textfile.txt”
Set FSO = CreateObject(“Scripting.FileSystemObject”)
Set TSO = FSO.OpenTextFile(MyFile, ForWriting, Create)
Dim strLineIn
Dim intCtr
intCtr = 0
Do While intCtr < 5
       strLineIn = streamIn.ReadLine
       intCtr = intCtr + 1 Loop
       Do While Not streamIn.AtEndOfStream
strLineIn = streamIn.ReadLine
       TSO.Write “Data Read From Pipe: “ & strLineIn
       & vbCrLf Loop
TSO.Close
Set TSO = Nothing
Set FSO
```

The command to run this example is as follows:

```
pkzipc -console -silent=input,banner -recipient=PKWAREPartnerLinkCA@pkware.com
-cryptalgorithm=AES,256 -passphrase=pkwarepkware c:\download\test\sensitivedata.ZIP
 | cscript szSTDIN.vbs
```

The file `sensitivedata.zip` is an encrypted ZIP archive. The script `szSTDIN.vbs` is the VBScript program that is listed above. After this command runs, the contents of the output are in textfile.txt. Instead of writing to a file, the script may be changed to import directly into a database.

Figure 5, right, shows SecureZIP integrated into the round-trip communication between database and transfer applications. Data from the database application is piped to SecureZIP to be compressed and encrypted, and is then piped from SecureZIP to the transfer application. Similarly, inbound data is piped to SecureZIP by the transfer application to be decrypted and decompressed before being piped into the database.

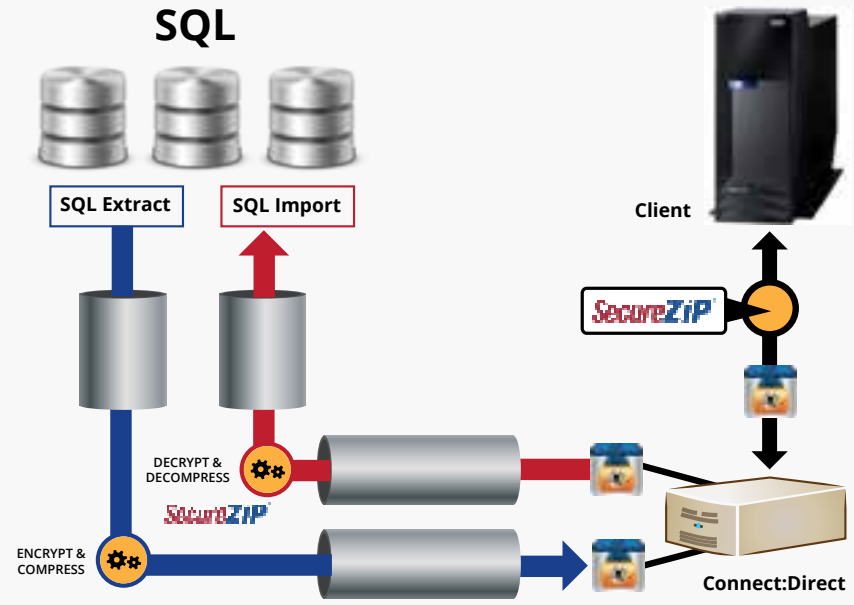


FIGURE 5—SQL AND TRANSFER APPLICATION WITH SECUREZIP FOR SERVER

Addressing Uncontrolled Encryption

Uncontrolled encryption can lead to an administrative nightmare and worse yet, jeopardize sensitive data security and compliance. Here are two prime strategies to prevent uncontrolled encryption.

POLICY MANAGEMENT

Policy management, or policy administration, is the means by which the organization can exercise oversight, control and proper use of strong encryption. This includes controlling how encryption is utilized and making sure that encrypted data can be recovered should a given encryption key (public and private) be forgotten, deleted or corrupted. Only with oversight of the users' encryption process will organizations be able to use encryption in a broad and effective enough way to address threats and regulatory compliance.

Auditors, regulators and customers all confirm that strong encryption is the "safe harbor" for sensitive data—even if the data is lost or stolen. Inevitably, security professionals have to grapple with the practical management of encryption in the context of the existing best practices. Certainly, sensitive data should be encrypted, but how can you address data oversight, recovery and audit/compliance?

The first layer of control is determining who in the organization has a legitimate encryption need. For example, backup administrators require access to files for backup purposes and to restore files, but do they require the need to decrypt the contents of the file? The organizational risk of encryption use can be greatly reduced by ensuring that the encryption application restricts access based on functions, roles, individuals and department. Several of the popular encryption and compression utilities fail to provide capabilities to manage access to encrypted data, leaving the power of encryption in the hands of many who are unprepared for its appropriate use. Policy management allows the organization to provide encryption to those workers to use with the appropriate control and oversight to satisfy auditors and regulators.

When used properly through policy administration, data-centric encryption prevents unauthorized access and tampering regardless of the state of data or where it travels.

CONTINGENCY KEY ACCESS

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 with approved allowances for decryption of data. The contingency key holders could be the InfoSec team from an organization-wide perspective, or departmental owners at a more granular level. Various business units or departments can have their own policy settings, and thus their own contingency keys. Policy management enforces that contingency keys are used effectively throughout the organization.

Using encryption without policy management and contingency keys can be dangerous and put 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.

The following story illustrates the risk that uncontrolled encryption presents. A manufacturing company in Germany asked if we could crack an encrypted ZIP file they provided us. A product manager had encrypted the designs for a new product and sent them to a competitor. We could tell that it was encrypted with AES-256 encryption, which employees had access to on their desktops/laptops. We explained to the company that this file was encrypted using strong security – but also without any contingency key administered by policy. Without the password, there was no way to decrypt the encrypted file. They could tell from the file name this was probably rogue behavior, but without the ability to prove what was in the file, they were not able to pursue criminal charges. Worse yet, having this information in the hands of their competitor was very damaging.

Many organizations use key escrow to allow authorized individuals access to decryption keys should they need to access encrypted data. This frequently happens when data discovery tools like Data Loss Prevention (DLP) are used to identify sensitive data that may leave the enterprise. Once identified, organizations can

understand where the sensitive data resides, where it is going and who has access. In many cases, the sensitive data is encrypted. A contingency key allows access to the data prior to it leaving the organization and eliminates the need for key escrow, since data can be accessed with the single master key.

Pairing policy management with alternate configuration files allows data-centric security to be effective and enforced. For example, certain jobs might need to encrypt with Federal Information Processing Standards 140-2 compliant crypto. Those jobs could be enforced to use FIPS 140-2 compliant data encryption, while other users might be required to use an AES-256 bit encryption algorithm. Different policies, configurations and contingency keys can be deployed for different groups of jobs or servers depending on the business and regulatory requirements.

Focusing on data protection really means implementing controlled encryption, for 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. An enterprise security product like SecureZIP for Server provides the ability to centrally manage encryption through policy and configuration files so that one or more contingency keys and various encryption standards can be applied to every encryption operation. This gives an organization peace of mind in how encryption is being used and the ability to access encrypted data for audit or recovery purposes.

Based on independent research by the Ponemon Institute, a 2013 multi-national survey concluded that "provisioning and access policy management" is the most important endpoint management feature. In addition, Osterman Research, an independent analyst specializing in workforce security and processes, reported in a survey that adoption of policy-based, automatic encryption increased from 27% in 2012 to 35% in 2013. Adoption of policy management that is tightly aligned with security strategies are definitely on the rise as security minded organizations look to increase protection of their critical data assets.

Going Beyond Compliance

If an organization handles sensitive data, it has to adhere to industry and government compliance regulations around data security. These mandates require that sensitive data be protected while in motion and at rest and that the organization has control and access to data as described above. Some compliance requirements are more specific in their guidelines.

For example, the Health Information Technology for Economic and Clinical Health (HITECH) Act requires organizations to prove data existed in its current format at a particular point in time. Using SecureZIP, an organization can verify that a file signed with a certificate in the past was created with a valid certificate, regardless of the current state of that certificate.

A variety of factors such as risk appetite, resources and regulatory demands pit companies in a race where they sometimes overlook the greater security threats. Are you trying to be very secure or are you focused merely on a lower threshold, like compliance? Security is very gray and complicated. Balancing your security philosophy should, at its core, mean you can out pace the hackers as well as the auditors.

If organizations fear the auditor more than they fear the bad guys, then the organization's data may be at risk of emerging threats. Worse yet, the bad guys know the regulations and the vulnerable areas not covered by regulation. Companies that fear the auditor simply work to pass the audit. Companies that fear the bad guy look for all the areas of vulnerability by hiring third-party penetration testing (Pen-Testing) to do both black box testing (where the pen tester can't see the underlying code) and white box testing (where the pen tester can see the underlying code).

There is a common myth that regulation and compliance benefits data security. In other words, thanks to these mandates audited by Qualified Security Assessors, data is more secure where it otherwise would not be. That leads some organizations in compliance to believe that they are also secure. This is not necessarily true. It only means that they are following the regulation. Compliance and security are not synonymous.

So, how could a regulation aimed at data security actually make companies who comply with it less secure? It happens when the compliance benchmark is seen as the main goal of data protection. Here, regulation sets a basement for security, which lowers the bar for security rather than raising it. Since the regulation only focuses on the minimum amount of security required to enforce the regulation across all companies, it in fact promotes the lowest common denominator.

Compliance with any standard does not equate to an assessment whereby a company's security is automatically appropriate. Standards are not necessarily commensurate with the size and complexity of the business environment or the type and amount of data involved. Security measures should go beyond the well-intended parameters of required mandates.

There are numerous examples of organizations that were in compliance with a regulation but still suffered a security breach. The most notable example is Heartland Payment Systems. They were found to be in PCI compliance yet lost millions of credit card data records because they were not secure enough. Deemed the largest credit card crime of all time, for months hackers had broken into Heartland computers used to process 100 million transactions from more than 175,000 merchants. Card issuers flagged suspicious transactions that revealed an overarching scheme to steal more than 130 million credit/debit card numbers as well as personally identifying information (PII). The hacker had breached Heartland a year before it was discovered, initially through an SQL injection attack. That then allowed the hacker access to all internal systems whereby the hacker was now acting as an insider with access to the underbelly of the sensitive systems.

Heartland has paid out millions to settle claims over the breach. As far as the data security ramifications, a post-mortem of the breach resulted in changes to PCI-DSS policies, as well as a move by Heartland toward a holistic, data-centric security approach.

ENCRYPTION AS A STANDARD: FIPS 140-2

Most U.S. government agencies and the private sector companies that work with them, such as banks and health-care insurers, are required to encrypt data with software that meets 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 be FIPS 140-2 compliant.

SecureZIP Server 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, other encryption tools are FIPS 197 compliant, which is not 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. 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.

SUMMARY

Enterprise security is really about defense in depth. The various layers of security must protect data at rest and as it moves outside of the enterprise perimeter. Today's security reality is that data is moving from many different endpoints and across many different platforms, making it impossible to protect every data endpoint in a consistent, comprehensive manner. By focusing on protection of the data itself, the dependency on endpoint protection is less critical and risks are reduced.

With a data-centric approach organizations can protect data while maintaining control and access in the face of today's evolving threats.

PKWARE®

CORPORATE HEADQUARTERS
201 E. Pittsburgh Ave.
Suite 400
Milwaukee, WI 53204 USA
1.800.219.7290

UK / EMEA
Building 3 Chiswick Park Chiswick High Road,
London W4 5YA
United Kingdom
+44 (0) 208 899 6060