

APPLICATION-LEVEL DATA PROTECTION

Implications, Challenges,
and Simplification



TABLE OF CONTENTS

- INTRODUCTION 4
- Protecting Data at Rest 4
- Data Protection-More than just Encryption 5
- The Complexity of Application-Level Data Protection 6
- Simplifying Application-Level Data Protection 7
- Abstracting How Data Is Protected from the Application 8
- Summary10

Introduction

We live in a digital era, where more data is being created by more applications than at any point in history. And data itself is more valuable than ever before, so it must be protected. This is no small task for business leaders, who continue to deal with the aggressive pace at which computer systems have become interconnected, distributed, and cloud deployed. Market pressures have made offering always-on, available-everywhere access to data and applications a baseline requirement, often with unfortunate side effects that leave sensitive data inadvertently exposed, whether by breach, insider malice, or simple human error.

We have seen industries and regulatory bodies establish standards by which data should be protected in an effort to stem the damage caused by the exposure of sensitive information. This, along with a plethora of practical business concerns, has led more and more companies to establish a comprehensive data protection strategy. It is safe to say that there has never been a more important point in history for protecting sensitive information, but are all data protection approaches equal?

Protecting Data at Rest

Most traditional data protection solutions focus on protecting data at rest by applying encryption techniques to sensitive data stored in disks, databases, or files. Cryptographic keys, leveraging these encryption techniques, are used to morph valuable sensitive information into meaningless digital blobs. Without the cryptographic keys to 'unlock' or decrypt the data, there is little threat from data exposure – be it intended or unintended – to stored data. It should go without saying that the secrecy of these keys and the ability to recall the keys to decrypt the data for some useful future business purpose when the time is right is of utmost importance.

Typically, encrypted data-at-rest is held in that secured state until an authorized application requests the data, at which point the data is decrypted and handed off to the application in its original form (often referred to as 'in the clear'), rendering previously secured data no longer protected once it leaves storage. This approach to data protection might be sufficient if the data never left its secure storage. However, value can only really be derived from data when it is in use – being handled, processed, analyzed, transferred, manipulated, etc. – by an application. One might argue that the whole point of storing data is to derive value from it at some future point, which almost always involves an application outside of secured storage, so any sensitive data worth storing should be worth securing wherever it is stored or used outside of storage too, to avoid risk of exposure.

With encrypting data-at-rest, one must keep in mind that the protection itself (the ability to secure or unsecure a piece of data) is applied at the storage level (disks, databases, files, etc.). There is no control over the data after it leaves these locations.

Data Protection – More than just Encryption

Before we move much further, let us examine for a moment the nuances between data encryption and protection. In addition to traditional encryption algorithms that leverage cryptographic keys to obfuscate data, there are a variety of other techniques to protect data, including tokenization, hashing, digitally signing, data redaction, and various other types of data masking. Collectively these approaches allow data to be secured to some level. However, a global movement to regulate the protection of personally identifiable information (PII) to address ongoing concerns around data privacy has also fundamentally expanded how one might define and facilitate data protection. In order to enforce data privacy, a data protection solution must be able to fundamentally define and enforce:

-  what data should be protected,
-  how protections should be applied,
-  who can access the data, and
-  what form data takes for each user when access is granted.

Data protection solutions must, of course, be able to leverage strong encryption and sound cryptographic key management to protect data, but they must also be able to deliver additional security measures such as tokenization and data masking, any of which might leverage a variety of techniques. Role-based access controls must be employed to not only restrict access to sensitive data, but also to intelligently manipulate data, such as applying redaction masks, to ensure that sensitive data is only exposed to those authorized and only to the extent required to meet some sanctioned business objective. Tracking, reporting, and alerting functionality that can support real-time and audit-based tracking and compliance are also important. Modern data protection is much more than just encryption.

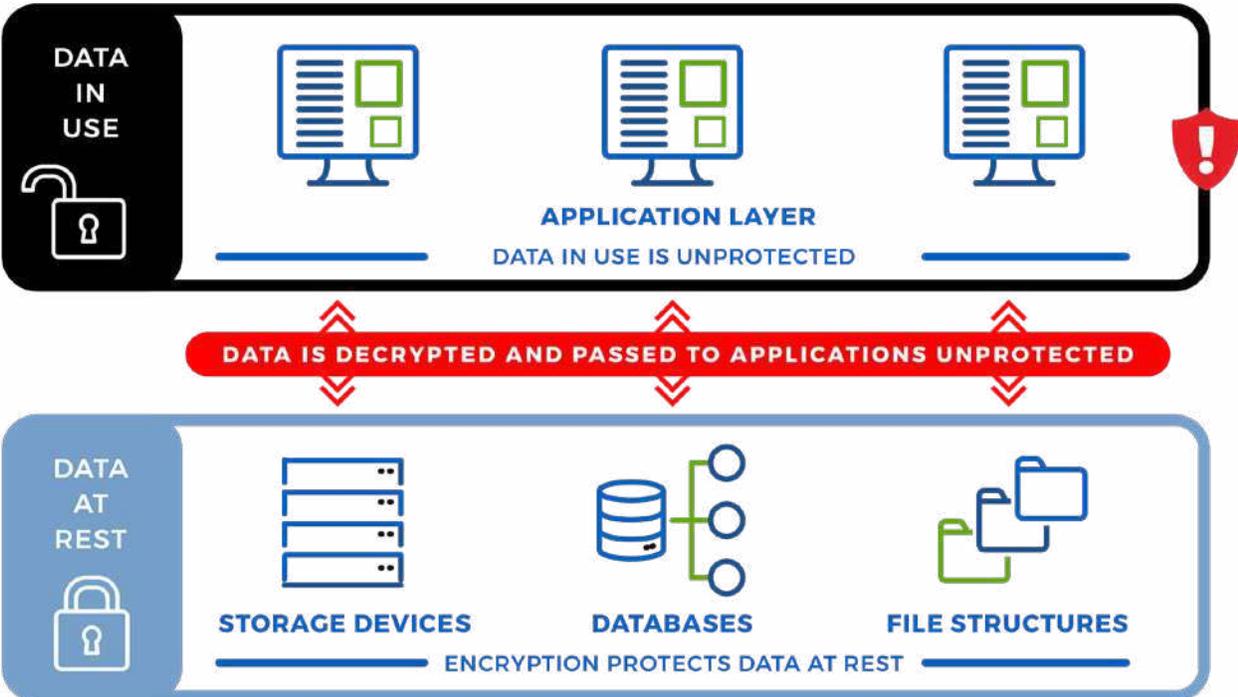
The fact of the matter is that traditional data-at-rest encryption solutions simply do not address protecting data in use. This is particularly concerning when we consider the fact that the vast majority of breaches are targeting the higher application layers of an enterprise, where data is exposed and more susceptible, not necessarily burglarizing data centers, stealing backup disks or databases. As the world of commerce grows more and more interconnected over time, it becomes clear that the most ideal place to protect sensitive data is right at the application at the moment it is created or imported into an enterprise, rendering data secure irrespective of where it is moved, used, or stored. Easy, right? Well, not so fast – application-level encryption has a reputation for being messy.

The Complexity of Application-Level Data Protection

The traditional approach to encrypting data in an application is to interweave cryptographic libraries and key management functionality into the application itself. The application then leverages this interwoven cryptography as an internal encryption engine to secure data in real-time within the application. This can be complex. To start, securely generating and managing cryptographic keys for application-native data encryption can be challenging enough. Add to this that applications must now also be able to define users and administer role-based data controls, define and enforce masking techniques, issue tokens, schedule and revoke keys, and the myriad of other elements necessary to enforce modern data protection and meet compliance standards, and it quickly becomes more challenging. Compound these challenges with the need for multiple applications to access the same encrypted data, perhaps across multiple operating systems and multiple cloud environments, and it becomes exponentially more complex and time consuming to implement.

These complexities make it challenging to implement modern data protection in general, but as different application development teams attempt to consistently enforce a shared data protection policy or policies within each of their various applications, it can also be exceptionally error-prone. Concerns over inconsistently implementing data protection policies across applications remains a top concern for most InfoSec professionals considering application-level data protection. The conclusion that enterprises have sometimes reached is that application-level data protection is just too complex to be practicable. It is no wonder that most businesses stick with securing the data at rest and leave it in the clear at the application layer! In order to protect data at the higher application layers where data in use is vulnerable, application-level data protection must be simplified.

DATA IS AT RISK AT THE APPLICATION LAYER

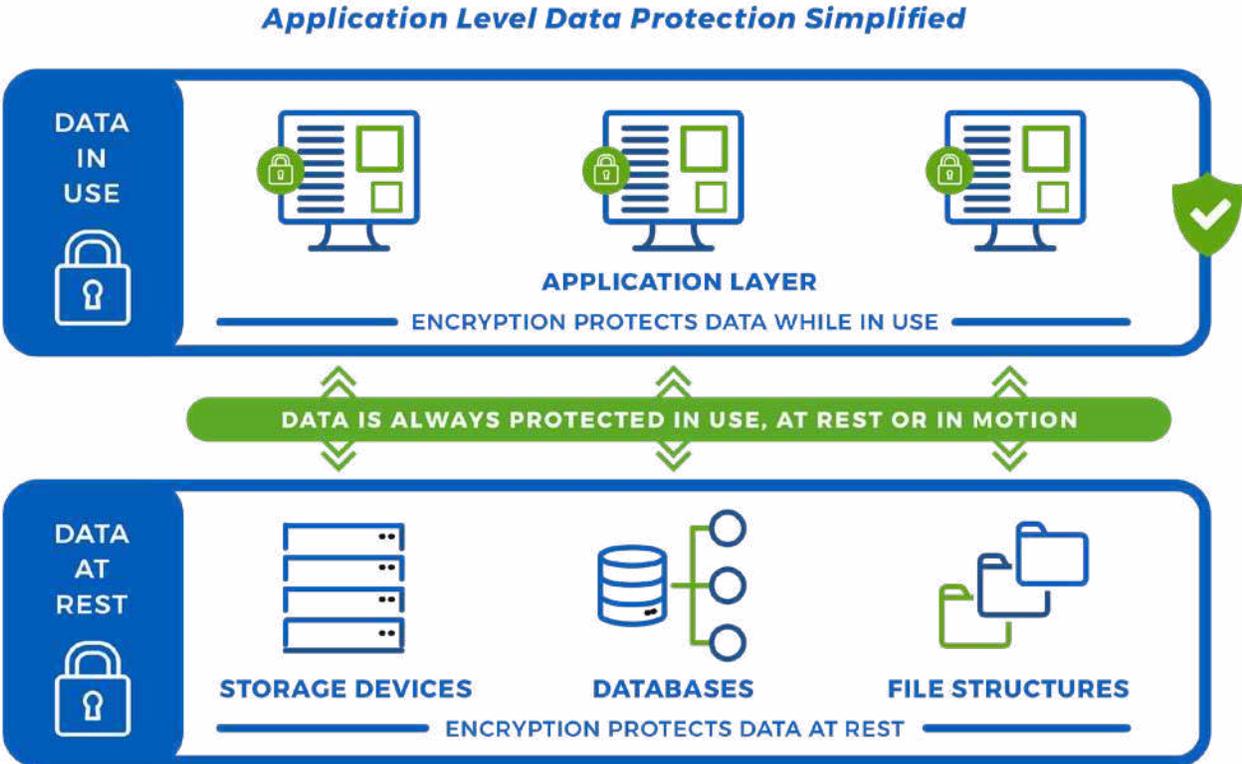


Most data is unprotected while in use.

Simplifying Application-Level Data Protection

In order to simplify application-level data protection, we must stop interweaving cryptography into applications and, instead, find ways to abstract how data is protected from the application itself. Data-centric security architectures must be employed to bind protection directly to data at the application level without relying on the application's internal functionality to actually protect the data. These changes in approach allow information security teams to develop data protection policies that cover all of the elements of modern data protection that can be applied to sensitive data at the application level, yet independently from any given application.

Approaching data protection in this manner inherently enables separations of duties - allowing InfoSec teams to establish policies in a centralized data protection policy engine and application programmers to implement these policies in a few lines of code without any real knowledge about the intricacies of the data protection policies themselves. Not requiring programmers to be cryptography experts is both drastically more efficient from a development perspective as well as safer from the threat of insider data exposure (accidental or otherwise), since programmers do not have intricate knowledge of the keys, algorithms, or other specifics used to protect data.



Protect data everywhere it is used, transported or stored!

Abstracting How Data is Protected from the Application

Envision a centrally managed data protection policy engine capable of orchestrating modern application-level data protection. The solution would need to be able to define and enforce all of the pertinent information necessary to comprehensively protect data, including:

How should data be protected?

- Define what encryption techniques should be used.
- Define how tokens are generated and used.
- Define hashing or digital signatures approaches.

How to manage cryptographic keys?

- Define the types of cryptographic keys required.
- Define how keys will be managed (assigned, restricted, rotated, etc.).
- Schedule keys to rotate and expire at certain times.

Who should be able to access data, and in what form?

- Define Users / User groups.
- Assign permissions that restrict users to authorized functions.
- Define any number of redaction masks to restrict data exposure.
- Correlate specific masks to specific user permissions to maximize protection.

How to migrate data protection posture over time?

- Set and track version controls to update policy information on the fly.
- Work with backward compatibility to unlock data secured under a prior version.
- Include customizable and flexible tracking, alerting, and reporting functionality.

Through simple API or CLI calls, an application could present a piece of data to the centralized data protection policy engine, and the security techniques used to secure the data (encryption, tokenization, hashing, signing, masking, etc.) would be applied to the data instantly, right at the application, in exact accordance with the Data Protection Policy. The application could then follow whatever data handling methodology it would normally follow, without needing to manage any of the data protection functionality internally. This secured data could then be passed to another application, transmitted to some endpoint, or placed into a storage location (disk, database, file structure, etc.), all with the confidence that the data has already been secured before it is moved, used, or stored.

When an application accessed a piece of secured data, the application could simply present the locked data to the centralized data policy engine, which would instantly 'unsecure' the sensitive data for the application, while exposing the data only to the level authorized for that specific user or group. For example, a sensitive card number might be completely exposed for a payment application to process an invoice, but the same data might be redacted except for the last four characters for a customer support organization to verify identity. This concept of dynamic data masking is a key ingredient in consistently enforcing data protection and privacy. When data protection is implemented at the application layer, we know that stored data is secured, transmitted data is secured, and application data is secured right up until the very moment it is exposed for authorized business functions.

This centralized data protection engine approach to application-level data protection also provides distinct advantages in the area of cryptographic key management. For most enterprises, securely generating and managing cryptographic keys remains a top challenge for application-level data protection. Without question, leveraging a certified general-purpose hardware security module (HSM) is a best practice for securely generating and protecting cryptographic keys. However, if the policy engine could connect directly to the HSM, the data protection engine could intelligently assign and manage cryptographic keys for the data defined in its data protection policies, without requiring key exchanges with an application or between multiple applications. It should be self-evident that the best way to reduce the complexity of sharing keys across multiple applications is to remove the need for applications to share keys altogether. This reduces complexity by allowing an InfoSec team to manage all cryptographic keys centrally, and it also removes the need for application programmers to understand, access, and handle keys at all – and added benefit to avoid inadvertently exposing keys or data in applications.

When a centralized data protection policy engine abstracts how data is protected in an application from the application itself, the net effect is a drastic simplification of application-level data protection. Instead of tremendous development time and costs that are prone to inconsistencies, application-native data protection can be accomplished in as little as three lines of code. Data protection can be centrally managed by data protection experts who define how data should be protected, and who can enforce a policy with perfect consistency across any application or data source. Data can always be protected at the moment it is created, before it is moved, used, or stored.



Learn how **EncryptRIGHT** from **Prime Factors** helps to simplify application-level data protection.



Learn how **nShield** from **Entrust** helps to establish a safe, trusted platform for cryptographic processes and key management.

www.primefactors.com

www.entrust.com