PROTECTING YOUR DATA WITH ENCRYPTION

Ed Leighton-Dick

Kingfisher TECHNOLOGIES

ABOUT ME

- Email: ed@leightondick.com
- Blog: edleightondick.com
- Twitter: @eleightondick

Kingfisher TECHNOLOGIES | I-380PASS | PASS | Microsoft CERTIFIED Solutions Expert Data Platform | MVP Microsoft® Most Valuable Professional

AGENDA

- Why should we use encryption?
- Three general types
  - In-Column
  - At-Rest
  - In-Transit
- Protecting Your Keys

Kingfisher

WHY SHOULD WE USE ENCRYPTION?

ZDNet

Cyberattacks caused the leak of one billion records in 2014

March 17, 2015

Summary: *According to IBM, roughly one billion personal records were leaked in 2014 due to online threats and cyberattacks.*

Figure 5. Vulnerability disclosures growth by year, 1996 through 2014

Source: IBM X-Force® Research and Development

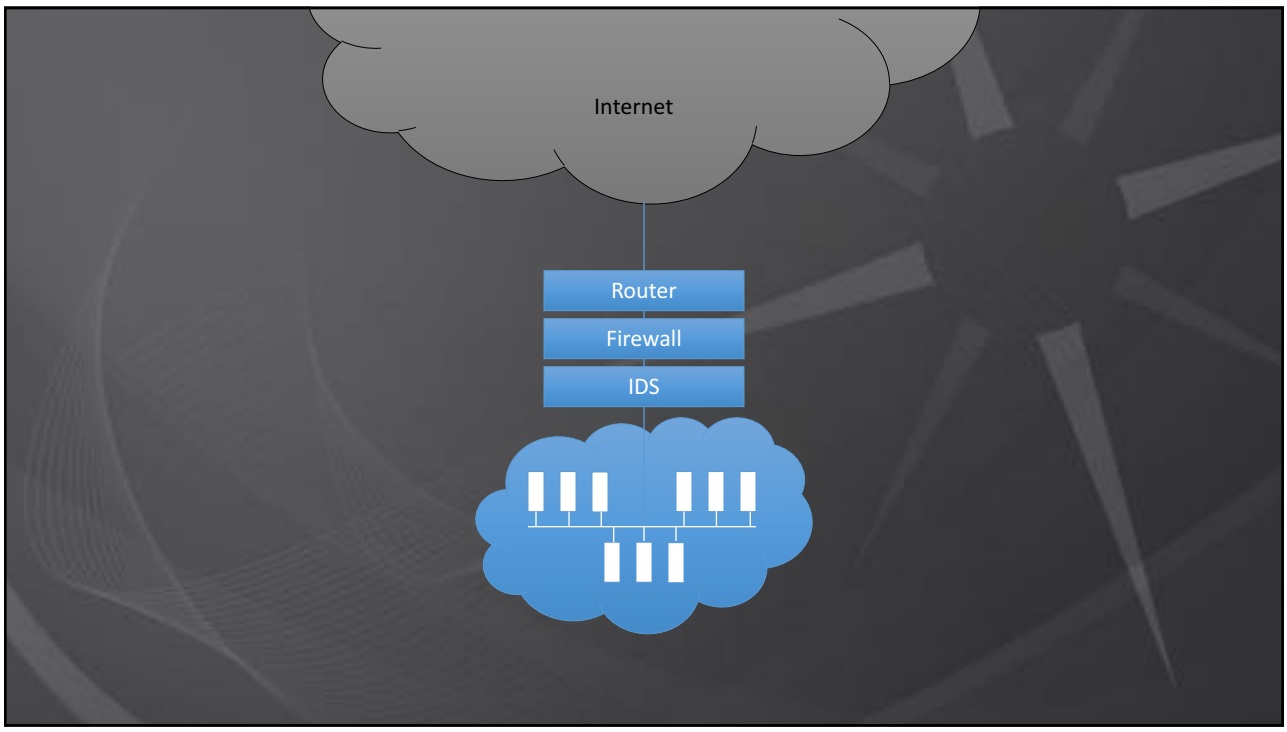
"In 2014, approximately one billion records of personally identifiable information (PII) were leaked online, according to IBM X-Force….The majority of these records were stolen from US companies."

"Anthem Inc. stored the Social Security numbers of 80 million customers without encrypting them, the result of what a person familiar with the matter described as a difficult balancing act between protecting the information and making it useful." (*The Wall Street Journal*, February 5, 2015)

WHY SHOULD WE USE ENCRYPTION?

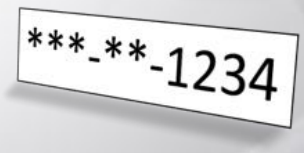
- Provides assurances to stakeholders
  - Customers, Employees, Management, Investors, etc.
- Satisfies regulatory requirements
- Provides a fail-safe for when other methods fail

Kingfisher

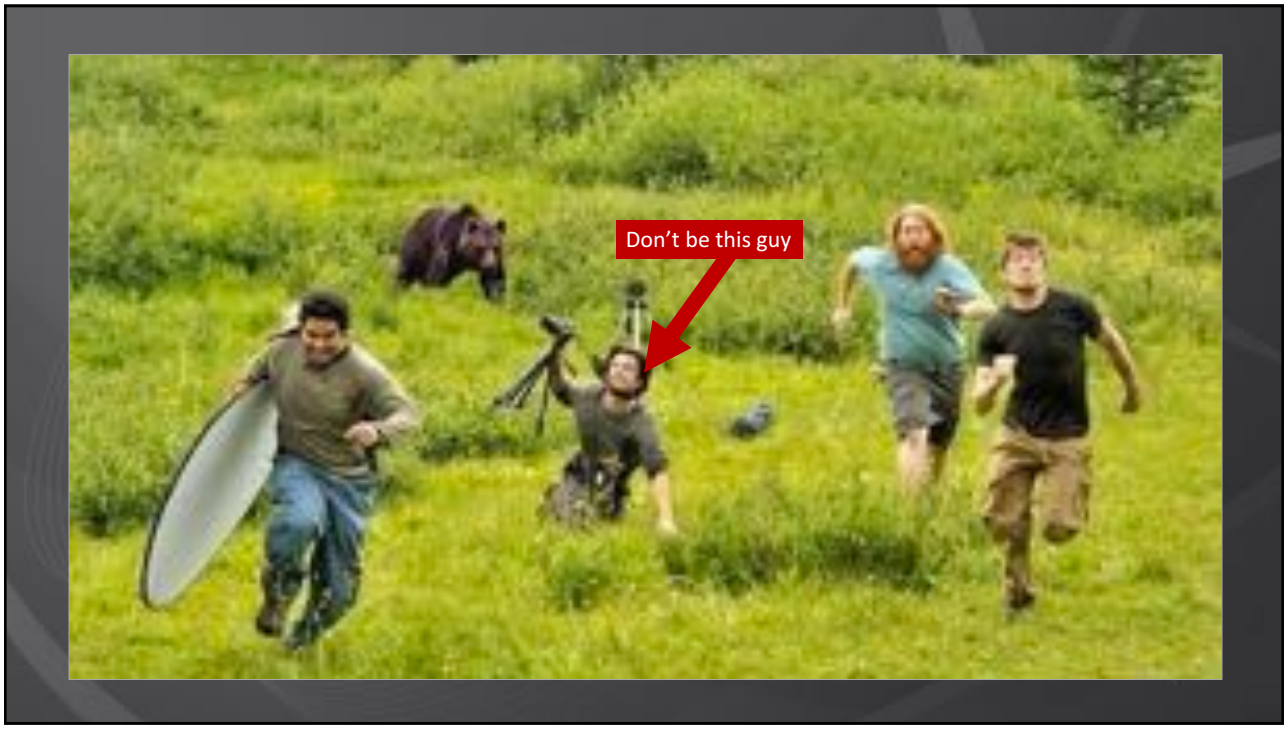


HOWEVER…

- Encryption protects against mass data theft
  - … which keeps the company off the front page
- All attacks aren't mitigated by encryption
  - Theft by authorized users
  - Social engineering attacks



Kingfisher



YOUR GOALS

1. Make your company a harder target than someone else
2. Make stolen data useless without expending significant resources (see #1)

Kingfisher

THIS SOUNDS HARD…

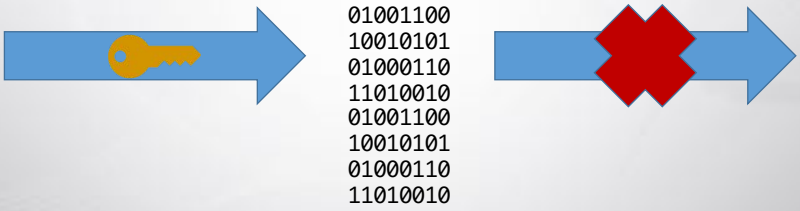
- … and it is!
- … but not as hard as you think
  - Built-in support
  - Day-to-day management is minimal

Kingfisher

IN-COLUMN ENCRYPTION

HASHES

TEXT



```
01001100
10010101
01000110
11010010
01001100
10010101
01000110
11010010
01001100
10010101
01000110
11010010
```

Kingfisher

HASHES

- Very fast
- One-way encryption
  - Encrypted phrase cannot be derived directly
- Commonly used for passwords or comparisons

Kingfisher

Hashes in SQL Server

- HASHBYTES
  - VARBINARY output
  - Algorithms
    - Use SHA2_256 or SHA2_512 when possible
    - 5 other algorithms available for compatibility
    - **WARNING:** Some older algorithms have been compromised!
      - Longer keys => Harder to crack

Kingfisher

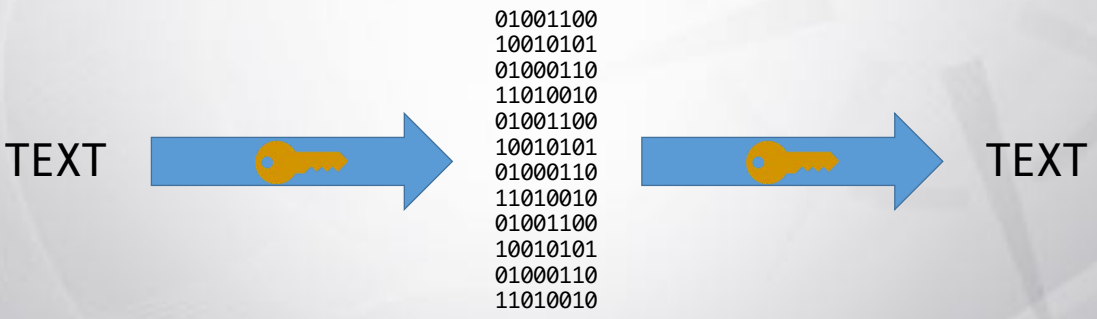
Hashes in SQL Server

- PWDENCRYPT/PWDCOMPARE
  - Deprecated
  - Uses that SQL Server version's password hashing algorithm
  - Commonly used to check for weak or blank passwords in sys.sql_logins

Kingfisher

DEMO: HASH ENCRYPTION

SYMMETRIC KEYS



Kingfisher

SYMMETRIC KEYS

- Fast
- Same key (password) encrypts and decrypts
- Commonly used for securing data
- Should be secured by another key
  - Can be encrypted by multiple protections at the same time
- Cannot be backed up
  - … But same password => same symmetric key

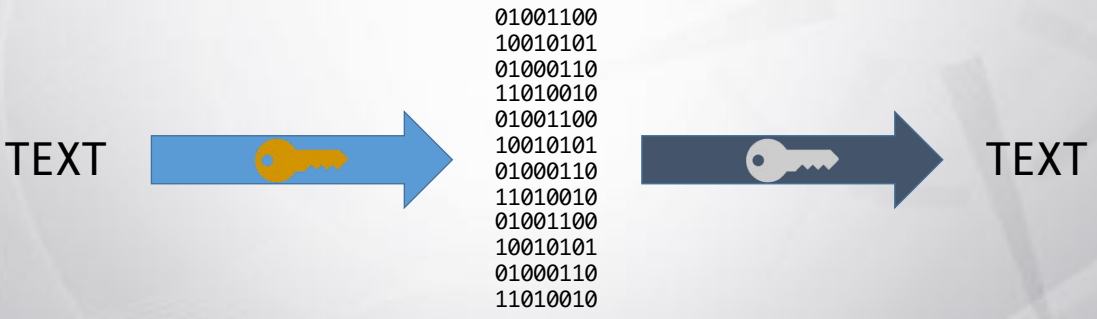
Kingfisher

Symmetric Keys in SQL Server

- CREATE SYMMETRIC KEY
  - Use AES algorithms when possible
  - RC4, RC4_128, DESX deprecated
- Encrypt with password or other key
- Cannot be backed up

Kingfisher

ASYMMETRIC KEYS



Kingfisher

ASYMMETRIC KEYS

- Slow
- Public/private key pair
  - Public key encrypts (Can be safely shared)
  - Private key decrypts
- Used to encrypt symmetric keys, sign data
  - Not recommended for securing data due to speed

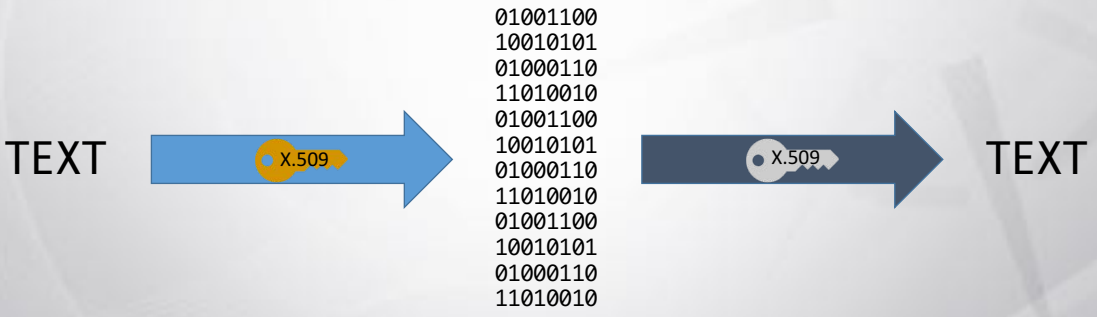
Kingfisher

Asymmetric Keys in SQL Server

- CREATE ASYMMETRIC KEY
  - Encrypt using
    - Password
    - Certificate
    - Asymmetric key
- Can be imported
  - .sn file (Strong Name utility)
  - Executable file
  - Assembly

Kingfisher

CERTIFICATES



Kingfisher

CERTIFICATES

- Specific type of asymmetric key
  - X.509v3 standard compliant
    - Code-signing ("Authenticode")
  - Has expiration date
  - Can be imported
  - Can be exported
- Uses
  - Encrypt symmetric keys
  - Secure and/or encrypt connections
  - Sign packages

Kingfisher

Certificates in SQL Server

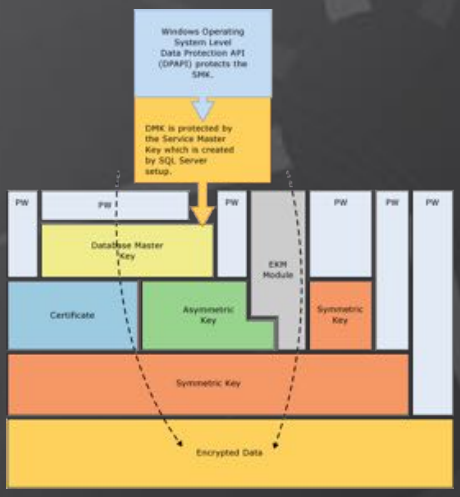
- CREATE CERTIFICATE
  - Encrypt using password (optional)
    - Prevents automatic decryption by master key
- Can be imported
  - .cer file (makecert utility)
  - .sn file (Strong Name utility)
  - Assembly
  - Authenticode certificate from third-party
- Can be created with or without private key
- Expiration date not checked by encryption, signing functions

Kingfisher

DEMO: IN-COLUMN ENCRYPTION

SQL SERVER'S ENCRYPTION HIERARCHY

Image Source: MSDN
(http://bit.ly/1EXdgLY)



DATABASE MASTER KEY

- Can be encrypted by multiple passwords
- Automatically decrypted
- Used to encrypt keys in database
  - Symmetric keys
  - Asymmetric keys
  - Certificates

Kingfisher

SERVICE MASTER KEY

- Stored in master database
- Automatically created when first required
  - Derived from SQL Server service account
- Cannot be directly changed
  - Can be regenerated
  - Can be restored to a different machine
- Used to encrypt database master keys

Kingfisher

ALWAYS ENCRYPTED ARCHITECTURE

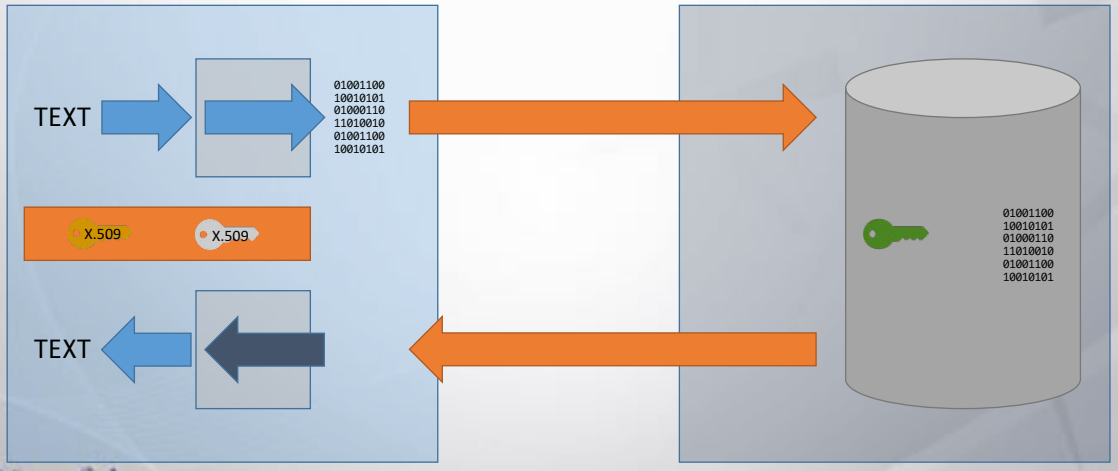


Image credit: MSDN (http://bit.ly/1Se0sm0)

Kingfisher

ALWAYS ENCRYPTED

- Encryption occurs in the data driver
  - ADO.NET, ODBC, JDBC drivers currently available
- Data is only unencrypted between application and driver
- SQL Server cannot decrypt data on its own
- Transparent to applications
- Searchable while encrypted
- Works on both on-premises and cloud databases
  - Available in SQL 2016, Azure SQL DB v12

Kingfisher

ALWAYS ENCRYPTED ARCHITECTURE

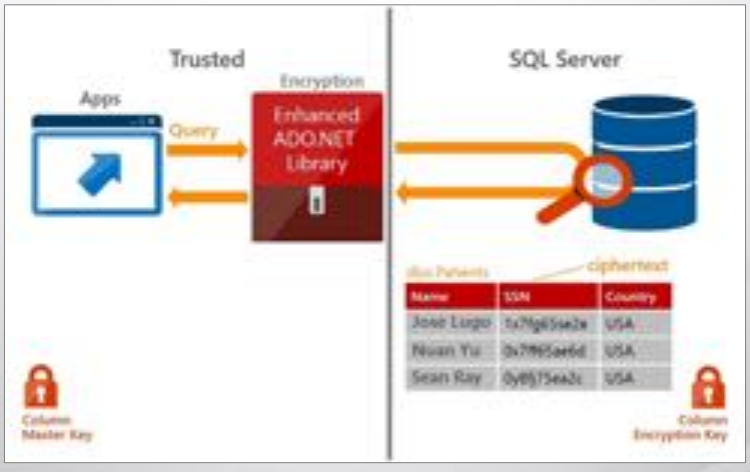


Image credit: MSDN (http://bit.ly/1Se0sm0)

Kingfisher

ALWAYS ENCRYPTED SOLVES SOME BIG PROBLEMS

- Strong encryption
- Separation of duties
- Consistent, simple implementation
- Encryption methods are not visible in code or database

Kingfisher

DEMO: ALWAYS ENCRYPTED

ALWAYS ENCRYPTED LIMITATIONS

- Comparisons
  - Deterministic – Equality only
  - Randomized – None
- Other operations disallowed
- What can be encrypted
- Performance
- Editions

Kingfisher

AT-REST ENCRYPTION

BACKUP ENCRYPTION

- Introduced with SQL Server 2014
  - Added to SQL 2012 SP1 CU4
  - Older versions cannot read encrypted backups
  - Backup to Windows Azure Tool for older versions (TechNet: http://bit.ly/1CgZ5xE)

Kingfisher

BACKUP ENCRYPTION

- Requires valid certificate in master
  - Used to create/secure symmetric key
- Original certificate must be present when restoring

Kingfisher

DEMO: BACKUP ENCRYPTION

TRANSPARENT DATA ENCRYPTION (TDE)

- What it does
  - Protects data on disk
  - No application changes required
- What it doesn't do
  - Protect against pages being read from cache
  - Protect against queries by authorized users

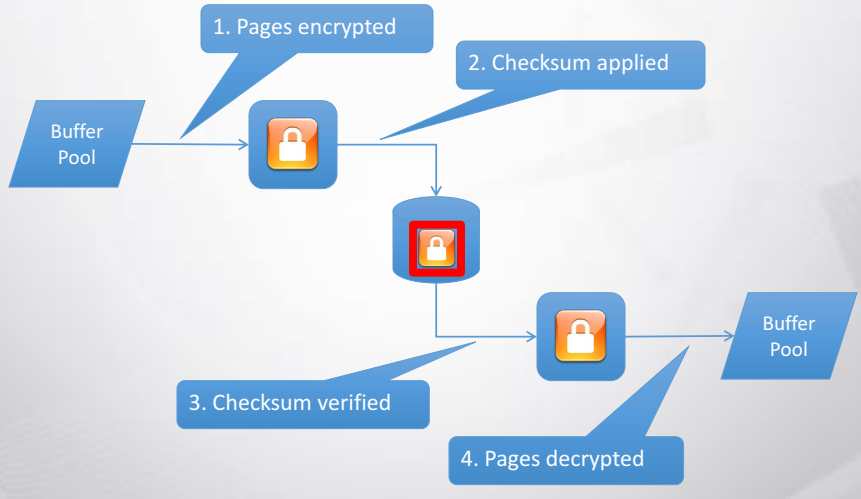
Kingfisher

TRANSPARENT DATA ENCRYPTION (TDE)

- Requires valid certificate in master
  - Used to create/secure symmetric key ("database encryption key")
- Certificate used to encrypt database must be present to restore database

Kingfisher

Transparent Data Encryption (TDE)



Kingfisher

TRANSPARENT DATA ENCRYPTION (TDE)

- Pages are encrypted when flushed to disk
  - Existing pages are encrypted in the background
- Pages are decrypted when read
- Side effects
  - Transaction logs are encrypted
  - TEMPDB is encrypted
  - Backups are encrypted

Kingfisher

LIMITATIONS OF TDE

- Enterprise Edition only
- Small impact on CPU utilization
- Instant File Initialization is not used
- FILESTREAM is not encrypted

Kingfisher

Monitoring TDE

- sys.dm_database_encryption_keys
  - Status codes (MSDN - http://bit.ly/1ITSyPM)
    - 0 = No key; unencrypted
    - 1 = Unencrypted
    - 2 = Encryption in progress
    - 3 = Encrypted
    - 4 = Key change in progress
    - 5 = Decryption in progress
    - 6 = Protection (certificate) change in progress

Kingfisher

DEMO: TRANSPARENT DATA ENCRYPTION

IN-TRANSIT ENCRYPTION

IN-TRANSIT ENCRYPTION

- SSL available natively
- IPSec (VPN) generally recommended
  - Persistent tunnel => Less overhead

Kingfisher

PROTECTING YOUR KEYS

SONY

GIZMODO

Sony Kept Thousands of Passwords in a Folder Named "Password"

December 4, 2014

Sony Left Passwords, Code-Signing Keys Virtually Unprotected

eWEEK

Researchers reportedly find hundreds of code-signing keys, at least one protected by an obvious password, while others find a folder labeled 'Passwords.'

The hackers that compromised Sony Pictures Entertainment's network leaked extremely sensitive information-technology credentials, including a collection of passwords and hundreds of certificates, one of which was used to sign the same malware used by the attackers to compromise the company.

December 10, 2014

PROTECT YOUR KEYS!

- Lost keys = Lost data
- Any encryption can be "broken" if the attacker gets the key!

Kingfisher

PROTECTING YOUR KEYS

- Always backup keys to a secure location
  - **No unnecessary access!**
- **Always backup master keys!**
- Always use a strong password to secure key files
  - Protect the password
- Use EKM, if available

Kingfisher

EXTENSIBLE KEY MANAGEMENT (EKM)

- Appliances that generate, store, and manage keys
- No one person has unfettered access
- Used for
  - Separation of duties
  - Simplified key management
- Cloud
  - Azure Key Vault
  - Townsend Security, SafeNet
  - Amazon (not compatible with SQL Server)

Kingfisher

REVIEW


THE DAILY BEAST
POLITICS ENTERTAINMENT WORLD U.S. NEWS TECH + HEALTH BEASTSTYLE WOMEN BOOKS
Photo Illustration by Emil Lendof/The Daily Beast
THEY CAN HEAR YOU NOW 02.20.15
Snowden's Revenge: New Mega-Spying Project Revealed
A giant cellphone surveillance program is just one of the dark NSA secrets being dragged out into the light, thanks to a certain whistleblower and a Russian cybersecurity firm.

REVIEW

- It is not possible with today's technology to completely eliminate the possibility of a successful breach
- Encryption mitigates intrusions by making stolen data useless
- Encryption is readily available…
  Use the tools at your disposal!

Kingfisher

QUESTIONS?

- Email: ed@leightondick.com
- Blog: edleightondick.com
- Twitter: @eleightondick

Kingfisher

THANK YOU FOR ATTENDING!

~ Please remember to fill out your comment cards ~

Kingfisher TECHNOLOGIES