



Database Applied Cryptography

Anti-forensics techniques/Data Protection

Case study: SQL Server 2022



Content



01_Intro

02_Database Applied Cryptography

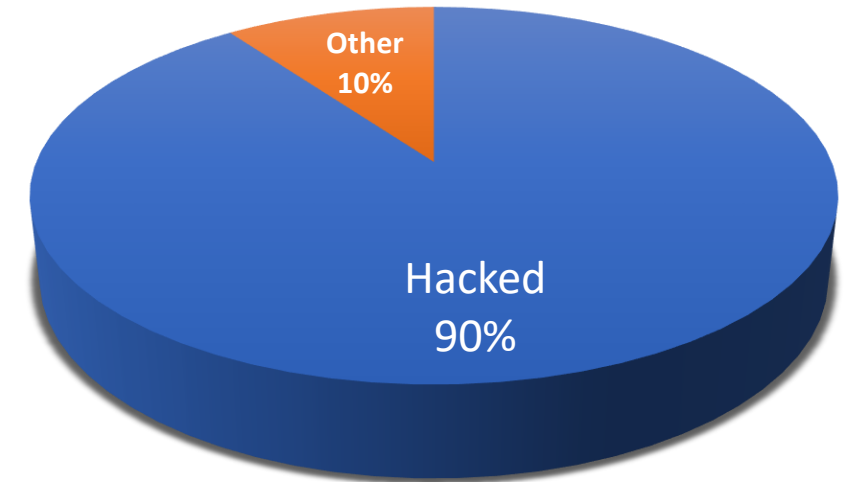


Intro



01_Intro

- Back Up files
- Data and Log files
- Oversized database permissions
- Copy/Paste from apps.
- Export options
- SQL injection
- Backdoors
- Phishing
- Social engineering
- Hacking
- etc.



Data stealing

■ Hacked ■ Other

Low level file access



```
mc [dba@Z]:/mnt/c/Users/dba/Desktop
/mnt/c/Users/dba/Desktop/NORTHWND.MDF 512944/2688K 18%
.).,0.).).0.....(.....(c.o.n.v.e.r.t.(s.m.a.l.l.i.n.t.,i.s.n.u.l.l.(c.o.n.v.e.r.t.(b.i.n.a.r.y.(2.),
.r.e.v.e.r.s.e.(s.u.b.s.t.r.i.n.g.(. [.r.e.f.k.e.y.s.] ,2.9.,2.).).),0.).).0.....).....(c.o.n.v.e.r.t.(
.s.m.a.l.l.i.n.t.,i.s.n.u.l.l.(c.o.n.v.e.r.t.(b.i.n.a.r.y.(2.),r.e.v.e.r.s.e.(s.u.b.s.t.r.i.n.g.(. [.r.e.f.k.e.y.s
.],.3.1.,2.).).),0.).).0.....:s.....CREATE VIEW syssegments (segment, name, status) AS
    SELECT 0, 'system' , 0 UNION
    SELECT 1, 'default' , 1 UNION
    SELECT 2, 'logsegment' , 0
.0....^t.....CREATE VIEW sysconstraints AS SELECT
    constid = convert(int, id),
    id = convert(int, parent_obj),
    colid = convert(smallint, info),
    spare1 = convert(tinyint, 0),
    status = convert(int,
        CASE xtype
            WHEN 'PK' THEN 1 WHEN 'UQ' THEN 2 WHEN 'F' THEN 3
            WHEN 'C' THEN 4 WHEN 'D' THEN 5 ELSE 0 END
        + CASE WHEN info != 0 -- CNST_COLUMN / CNST_TABLE
            THEN (16) ELSE (32) END
        + CASE WHEN (status & 16)!=0 -- CNST_CLINDEX
            THEN (512) ELSE 0 END
        + CASE WHEN (status & 32)!=0 -- CNST_NCLINDEX
            THEN (1024) ELSE 0 END
        + (2048) -- CNST_NOTDEFERRABLE
        + CASE WHEN (status & 256)!=0 -- CNST_DISABLE
            THEN (16384) ELSE 0 END
        + CASE WHEN (status & 512)!=0 -- CNST_ENABLE
            THEN (32767) ELSE 0 END
        + CASE WHEN (status & 4)!=0 -- CNST_NONAME
1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit
```

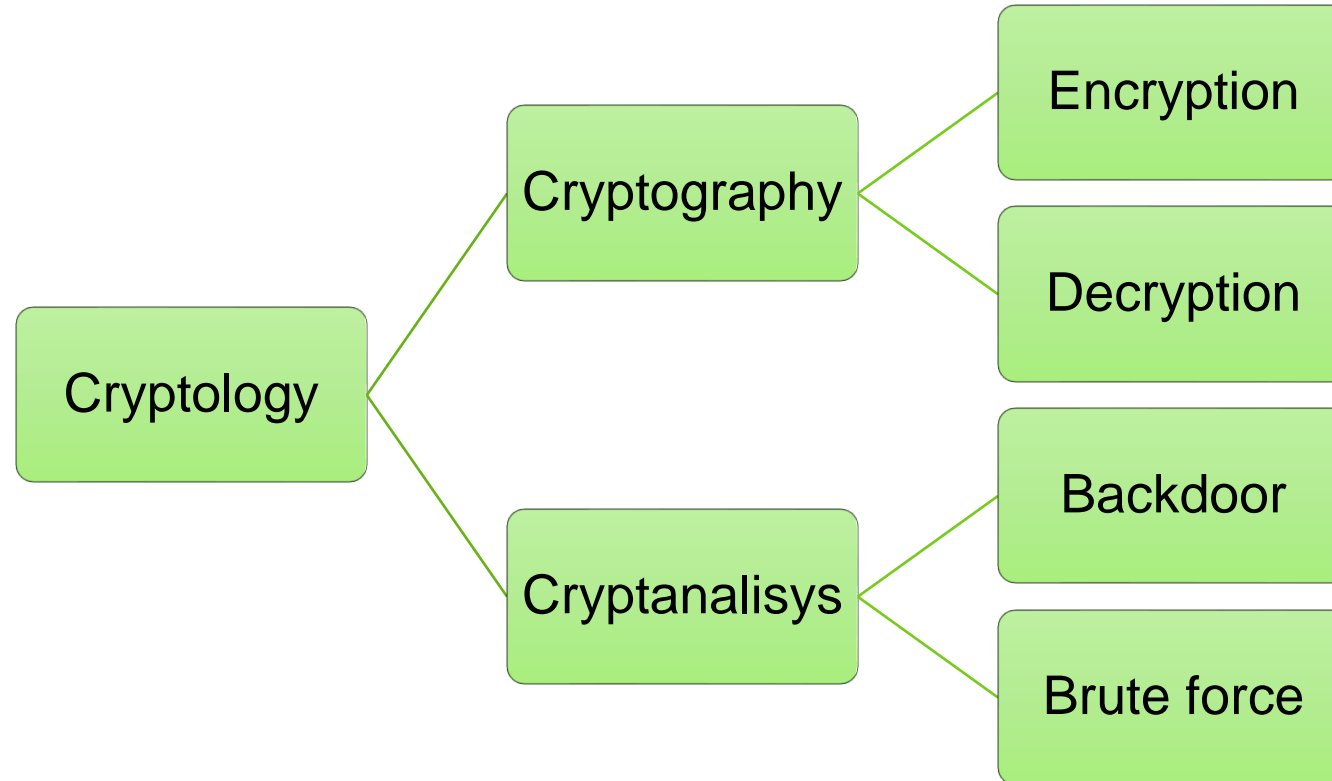
+

[illegible]



SQL Server Applied Cryptography

02_What is Cryptography



02_Types of Cryptography



- Symmetric cryptography
 - the sender and recipient share a key that is used to perform encryption and decryption
 - common symmetric algorithms are: Rijndael (AES) and Triple DES (3DES).
- Asymmetric cryptography
 - the sender encrypts data with one key, and the recipient uses another key for decryption
 - commonly used asymmetric algorithm is the RSA algorithm

02_SQL Server Applied Cryptography



SQL Server Encryption Hierarchy

SQL Server Encryption Algorithms

Pre-breach cryptography

- Symmetric key column encryption
- Always Encrypted
- Always Encrypted : Secure Enclaves

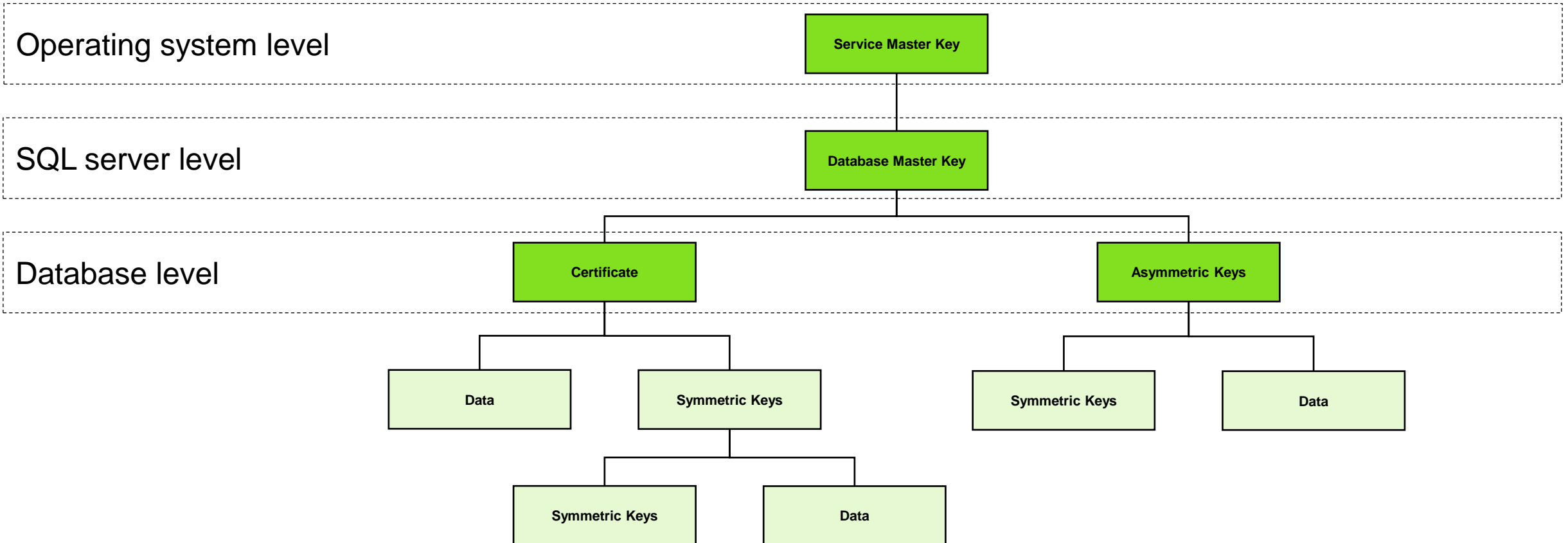
Post-breach cryptography

- Transparent Data Encryption
- Back Up encryption



254F1	21B2C809	8833B0CC
3ECAA	CB3EE	DE038D7F
2AA4D	04143	F571C83
7DED9	B57C	1820EE07
696DB	7D7F7	6DD29
0014D	41080	54E072
05552	534146D	8360929
18BFC	0F130429	90A60B99

02_SQL Server Encryption Hierarchy



02_SQL Server Encryption Algorithms



Beginning with SQL Server 2016 (13.x), all algorithms other than AES_128, AES_192, and AES_256 are deprecated. To use older algorithms (not recommended) you must set the database-to-database compatibility level 120 or lower.

Symmetric Encryption Algorithms

Keyword	Algorithm	Key Length (Bits)
AES_256	AES	256
AES_192	AES	192
AES_128	AES	128
TRIPLE_DES_3KEY	Triple DES (3-Key)	112

Asymmetric Algorithms

Keyword	Algorithm	Key Length (Bits)
RSA_2048	RSA	2048
RSA_1024	RSA	1024
RSA_512	RSA	512

02_Symmetric key column encryption



Symmetric Keys are created in a database, and they are always in that database

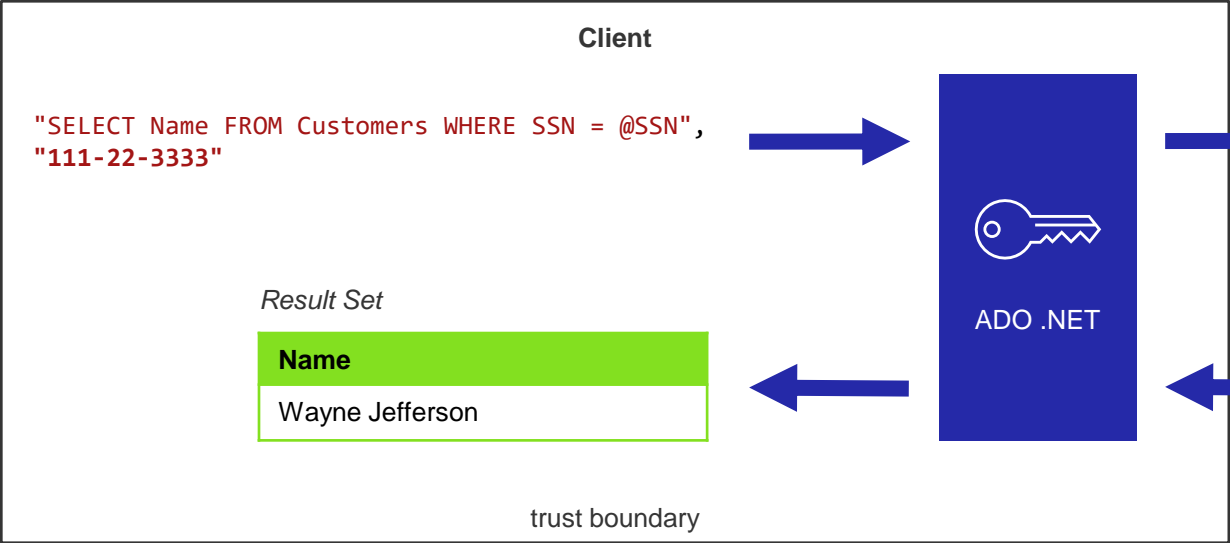
What can you encrypt/decrypt?

- Data in tables (column level)

Encryption require an additional CPU load from their use.

Results		Messages
	FirstName	
1	0x008EA7B31E68C4B8CA3307A5AA41CBCE010000007CD2971D4429634F322549A5563B3E90DB0B6388CEECB3A37533F1160CB043D9	
2	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000F2DCB70922F8B65231122AF2A4F9C41B919F704AFD75C0EE9D41325041D08C5851D8B311C5120BC7EDBE992A719670FA	
3	0x008EA7B31E68C4B8CA3307A5AA41CBCE0100000026A5F8C367E9C4524BFD880BA586457D65B4B7571B69D325E9AAB848CD4043C008A3D47A4454D3054ED6DBD6D861AB56	
4	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000B8BCD196FE4DB5124C6E3224FA2C0AC4715E053248B14DE399552082E1C864DA	
5	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000BC41B4D968C9233FB989558B4BCE6DD1553DBEBFF62E6E1EC01D76F818C8BEB71C9A23CAB8BC669A742738AE94F26D8E	
6	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000F071CFD8B17339A9528F5DD34599E5E3F48409D4B033A37F4C4AABFD8690D10CEC2A48F4F464F20ED49C91F40F92EFEA	
7	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000311B39CFB51BB66CC6669C62A44EE0F2C9DE3633DE51E5AD173F54FCD9C3CCB6432E523090662AEAD50716E8421BFD31	
8	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000CBFF8899034BFF048DC23C23C51D83720CAD717D456EDE5AF2546D4EBF44B591D2D19BF8E524C52FDFBDA6C83014F7EF	
9	0x008EA7B31E68C4B8CA3307A5AA41CBCE01000000F08B158296356D5BC2EE5EF151A8A7A40BBF7E032BDC5B220EE171411B35D48B712C0599491CD59395500D055517F8E1	
10	0x008EA7B31E68C4B8CA3307A5AA41CBCE0100000047EF032AA4ADC5314E7CC79D227DD5DC4B97134F028F6BAE3E5F763F7922B2898D6DF1E2A4F6B33451872DFFAF76947	

02_Always Encrypted - facts



Encrypted sensitive data and corresponding keys are never seen in plaintext in SQL Server

SQL Server or SQL Database

"SELECT Name FROM Customers WHERE SSN = @SSN",
0x7ff654ae6d

ciphertext

Result Set

Name
0x19ca706fbd9a



dbo.Customers

Name	SSN	Country
0x19ca706fbd9a	0x7ff654ae6d	USA

ciphertext

02_Always Encrypted - facts

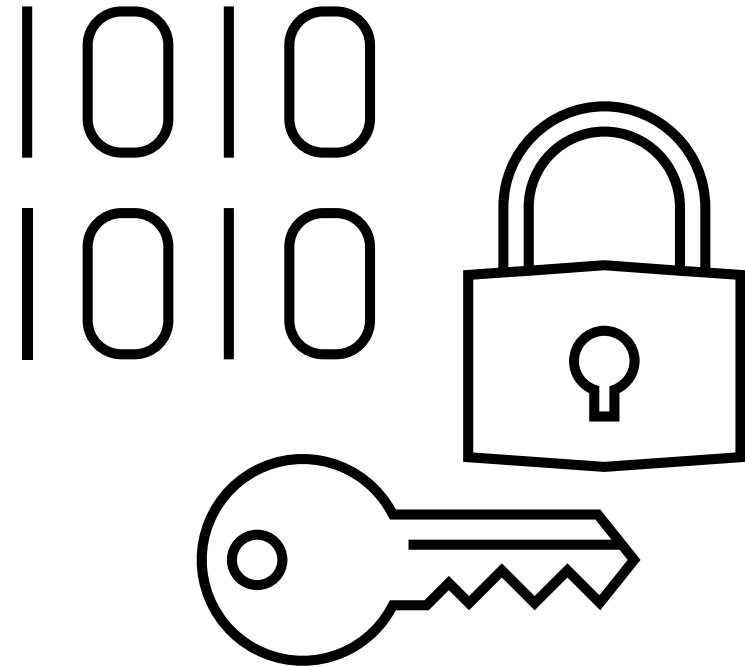


Randomized encryption

- Encrypt ('123-45-6789') = 0x17cfd50a
- Repeat: Encrypt ('123-45-6789') = 0x9b1fcf32
- Allows for transparent retrieval of encrypted data but NO operations
- More secure

Deterministic encryption

- Encrypt('123-45-6789') = 0x85a55d3f
- Repeat: Encrypt('123-45-6789') = 0x85a55d3f
- Allows for transparent retrieval of encrypted data AND equality comparison
- E.g., in WHERE clauses and joins, distinct, group by



02_Always Encrypted - Secure Enclaves

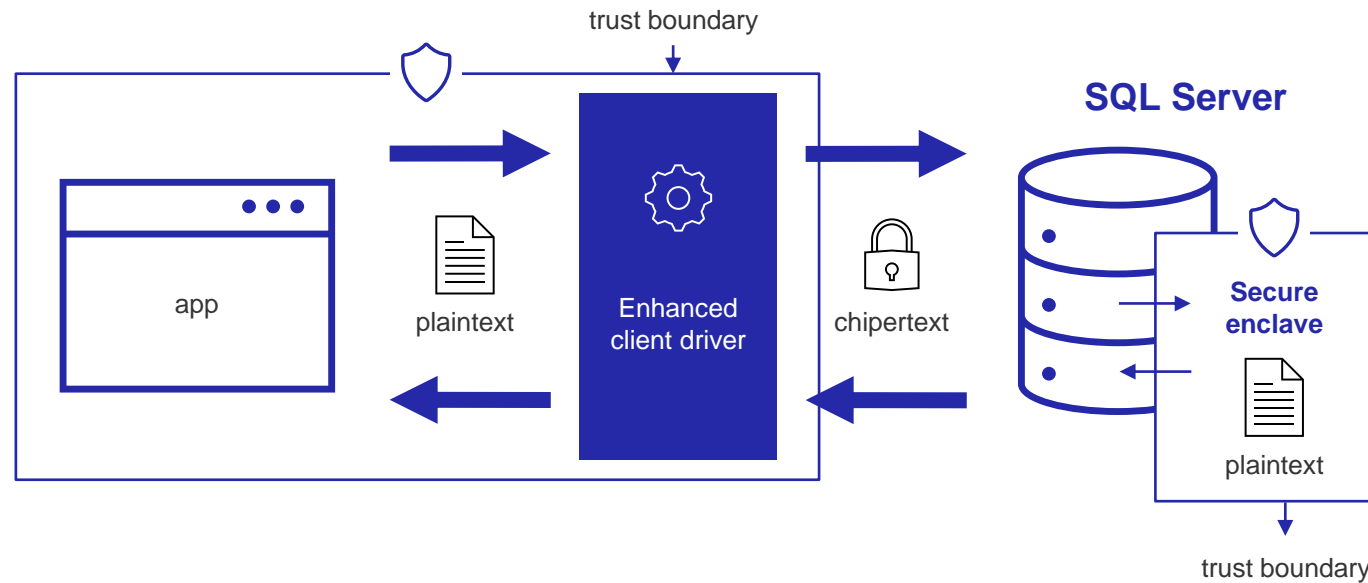


Current implementation : encrypt/decrypt at client

Great for security, but...
Rich computations impossible or expensive

Secure Enclaves

Protected, isolated area of memory (fast!) on the SQL Server machine; allowing...
Encrypt in place (e.g., key rotation), range queries, pattern matching – even on random



02_Transparent Data Encryption



Protects the data at rest by encrypting the data on disk.

The transaction log is encrypted

Backups are encrypted

Tempdb is encrypted for all operations.

Replication data is not encrypted

Filestream data is not encrypted

TDE does not provide encryption across communication channels



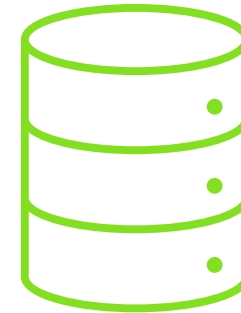
02_Backup encryption



Increase security of backups stored separate from the instance (another environment such as the Cloud)



Encryption keys can be stored on-premises while backup files in the cloud



Support non-encrypted databases (don't need to turn on TDE anymore)

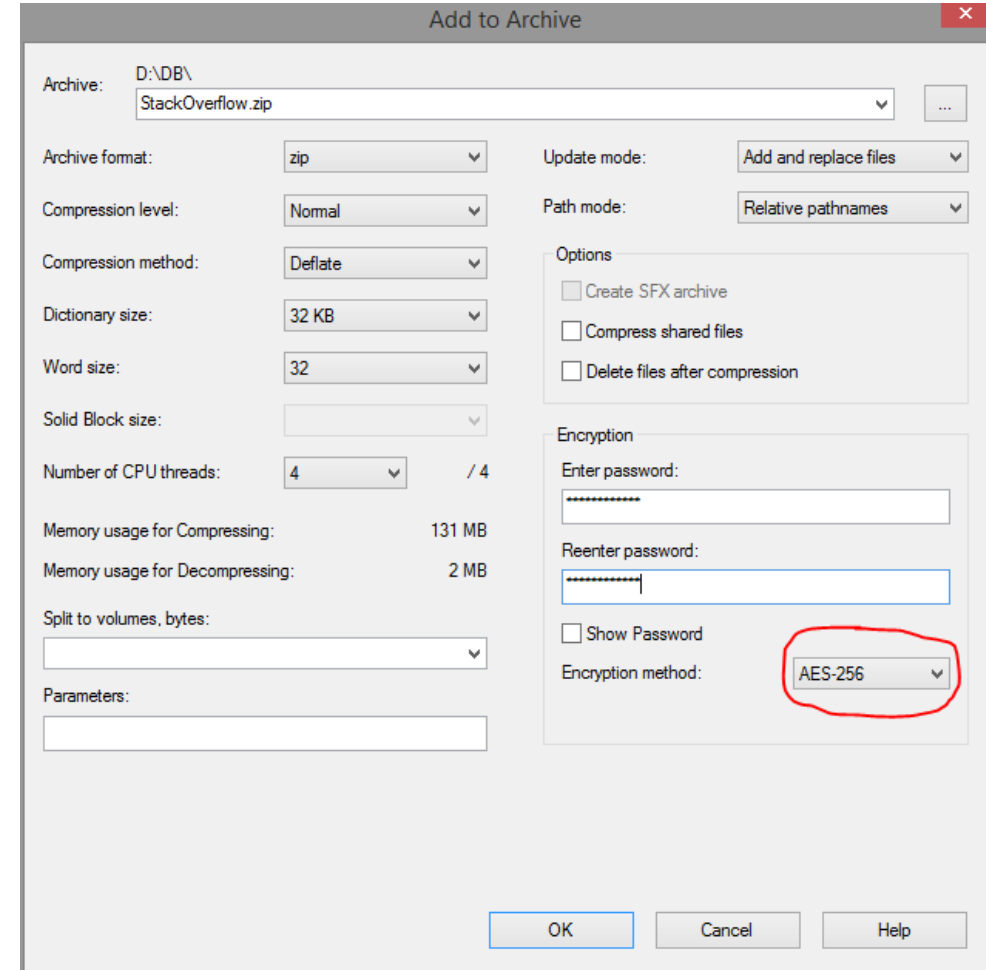
02_Better then nothing



7ZIP

WinRAR

Another tools



Demo section



- Column encryption symmetric key – **Live Demo**
- Column encryption asymmetric key – **Live Demo**
- Always encrypted – **3:15 min (Recording)**
- Transparent Database Encryption (TDE) – **2:20 min (Recording)**
- Backup encryption – **2 min (Recording)**

Additional resources



<https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/sql-server-encryption>

<https://github.com/jasminazemovic/Book-Securing-Sql-Server>

<https://www.sqlshack.com/sql-server-confidential-part-crypto-basics-sql-server-cryptographic-features/>

<https://www.sqlshack.com/sql-server-confidential-part-ii-sql-server-cryptographic-features/>



Time for questions