

Dynamic Data Masking

Dynamic Data Masking



Dynamic Data Masking

Dynamic Data Masking

ID	PersonName	EmailAddress	CreditCardNumber	SocialSecurityNumber
1	Anoop Kumar	abcdefgh@hotmail.com	1234-5678-4321-8765	123-45-6789
1	Rahul Gupta	amitguptaabcdefg@hotmail.com	8765-1234-5678-4321	231-45-6787
1	Amit Goel	amitgoelabcdefgh@hotmail.com	4321-1234-5678-4321	321-45-6700

ID	PersonName	EmailAddress	CreditCardNumber	SocialSecurityNumber
9590	AXXXr	aXXX@XXXX.com	xxxx-xxxx-xxxx-8765	xxxx
7604	RXXXa	aXXX@XXXX.com	xxxx-xxxx-xxxx-4321	xxxx
8453	AXXXI	aXXX@XXXX.com	xxxx-xxxx-xxxx-4321	xxxx

Random Number function – generates random number based on selected boundaries and actual data type. Can be applied only numbers, not string

Email function – Exposes the first letter and replace the domain with xxx.com

Default function – Full masking of data.
For numeric – 0
For String – XXXX characters

Credit Card function - Only the last four digits of Credit card are shown

Custom String function – You can define the exposed prefix, the padding string and exposed suffix



Dynamic Data Masking

- **Limit the exposure of sensitive data to non-privileged users**
 - You can decide the level of exposure of data
- **No change in physical layer**
 - Data in the database is not changed
 - Not the same as data encryption
- **No additional development effort needed at application level**
- **Security: Should not be used as a primary security layer**
 - Dynamic Data Masking should not be used as an isolated measure to fully secure sensitive data
 - ad-hoc query permissions can apply techniques to gain access to the actual data.
- **Other considerations**
 - Masked columns can be updated if user has permission
 - Export masked from source data results in masked data in target table



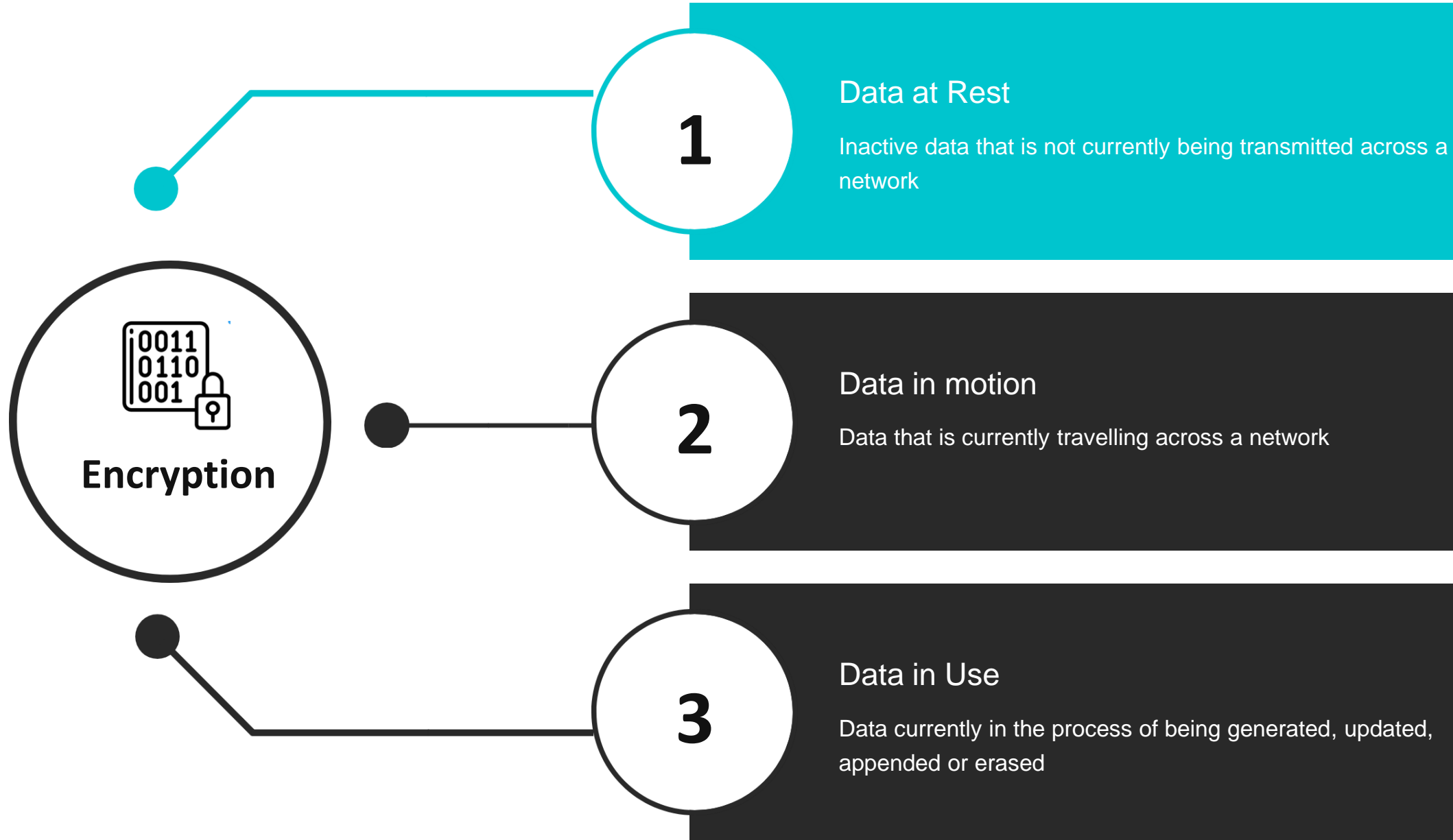
Dynamic Data Masking

- **How to Enable DDM?**
 - Portal
 - Powershell
 - `Get-AzSqlDatabaseDataMaskingRule`
 - `New-AzSqlDatabaseDataMaskingRule`
 - `Remove-AzSqlDatabaseDataMaskingRule`
 - `Set-AzSqlDatabaseDataMaskingRule`
 - Rest API

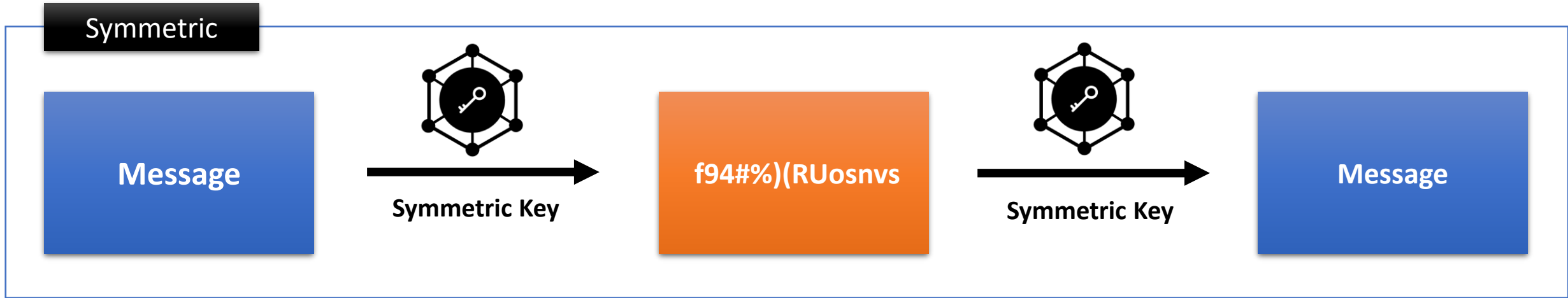


Encryption

Of data at rest, motion or use



Types of Encryption

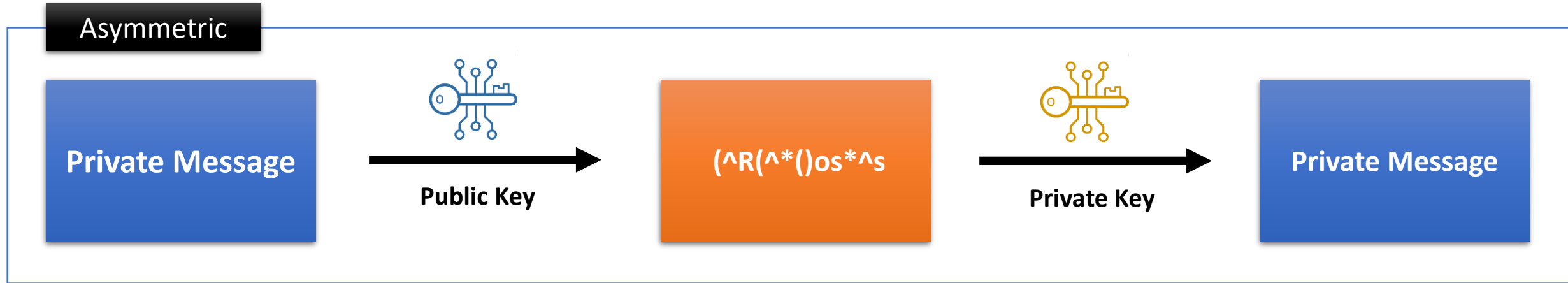


Highly performant



Secure key handling is difficult

Types of Encryption

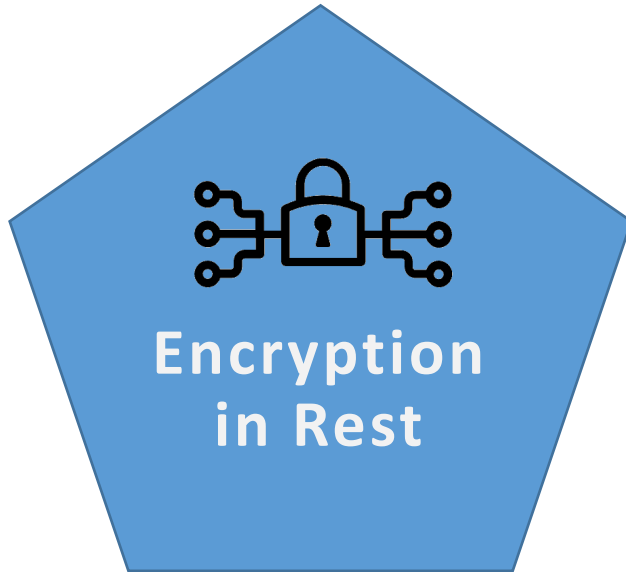


Set of keys (Public and Private) where one is use to encrypt and other to decrypt



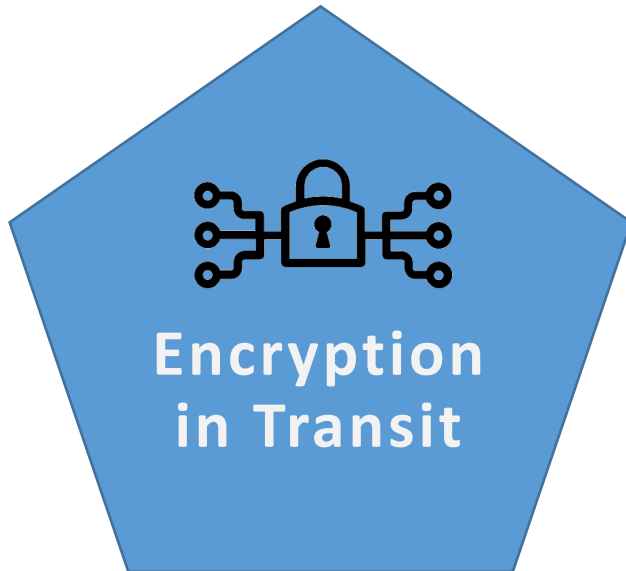
Computationally more expensive, but solves the key handling problem

Encryption in Rest



- By default all storage services encrypted data at rest
- 256-bit AES encryption
- Keys are stored in Azure Key Vault

Encryption in transit



- Any communication over the internet to Azure services is typically encrypted via SSL/TLS protocols
- Utilize site-to-site VPN or point-to-site VPN connections
- Or utilize ExpressRoute
- ExpressRoute communication is a private connection and not encrypted but workloads can be encrypted via SSL/TLS
- Storage services can be configured to require secure transfer



Types of Encryption

Deterministic encryption:

- This will always generate the same encrypted value for any plain text value.
- You can perform point lookups, equality joins, grouping and indexing on encrypted columns.

Randomized encryption:

- This is more secure than deterministic encryption because the encrypted value is generated in a less predictable manner.
- But you can't perform searching, grouping, indexing or joins on encrypted columns.



Types of Encryption

Column Encryption Key (CEK)

- Used to encrypt values in specific columns
- Encrypted versions of each CEK is stored in the database.

Column Master Key (CMK)

- Used to encrypt all the CEKs
- Must be stored externally in a secure key store
- Key store providers: Azure key vault, certificate store, HSM