New York Oracle Users Group, Inc.

NYOUG

Managing Risk: Understanding the New Options in Data Protection

Ulf Mattsson, CTO, Protegrity Corporation

NYOUG 25th Anniversary/2009

December 8, 2009

ISSA New York Metro Chapter - Information Security Systems Association

ISSA™ National Capital Chapter

Information Systems Security Association

The Global Voice

Ulf Mattsson, CTO, Protegrity Corporation

June 4, 2009

MasterCard Worldwide

How to Evaluate Encryption Technologies

Achieving PCI Compliance

ISACA® Serving IT Governance Professionals

New York Metropolitan Chapter

Metro Line

Volume 1, Issue 4 October 2009

Developing, Deploying and Managing a Risk-Adjusted Data Security Plan

Ulf Mattsson
CTO
Protegrity

Not too long ago, many security experts believed that the best way to defend data was to apply the strongest possible technological protections to all of the data, all of the time. While that plan may work perfectly in theory, in the real world of business this model creates unacceptable costs, performance and availability problems.

Managing Risk

How to Develop a Risk-adjusted Data Protection Plan:

1. Know Your Data & Find Your Data
2. Understand Your Enemy
3. Understand the New Options in Data Protection
4. Deploy Defenses
5. Crunch the Numbers

Step1: Know Your Data – Identify High Risk Data

- Begin by determining the risk profile of all relevant data collected and stored
  - Data that is resalable for a profit
  - Value of the information to your organization
  - Anticipated cost of its exposure

Data Field	Risk Level
Credit Card Number	25
Social Security Number	20
CVV	20
Customer Name	12
Secret Formula	10
Employee Name	9
Employee Health Record	6
Zip Code	3

Step 2: Find Your Data – Understand the Data Flow

Collection

- ‘Information in the wild’
  - Short lifecycle / High risk

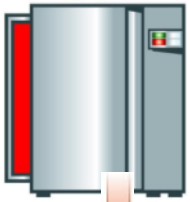
Aggregation

- Temporary information
  - Short lifecycle / High risk

Operations

- Operating information
  - Typically 1 or more year lifecycle
  - Broad and diverse computing and database environment

Analysis



- Decision making information
  - Typically multi-year lifecycle
  - Homogeneous computing environment
  - High volume database analysis

Archive

- Archive
  - Typically multi-year lifecycle
  - Preserving the ability to retrieve the data in the future is important

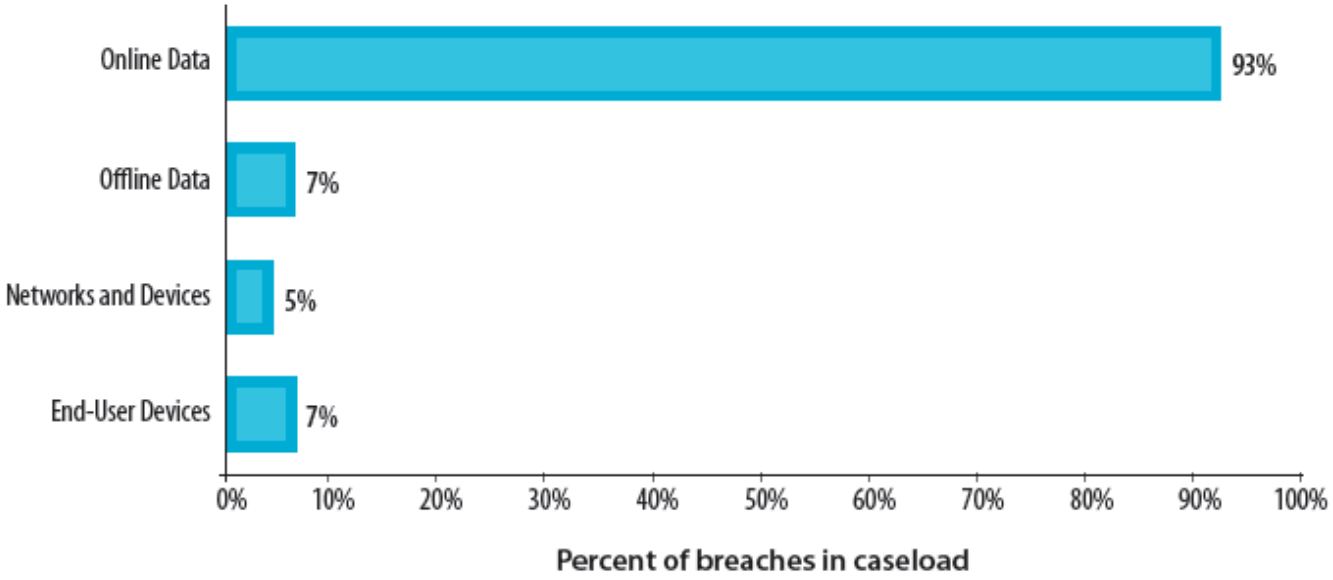
Where and When is Data Most at Risk?

protegrity

Online Data Under Attack – Not Laptops or Backup

Breaches attributed to insiders are much larger than those caused by outsiders

The type of asset compromised most frequently is online data:



87% of breaches could have been avoided through reasonable controls

Slide source: Verizon Business 2008 Data Breach Investigations Report

protegrity

Step 3: Understand Your Enemy – Probability of Attacks

Higher Probability

What is the Probability of Different Attacks on Data?

Errors and Omissions

Lost Backups, In Transit

Application User (e.g. SQL Injection)

SQL Users

Network or Application/RAM Sniffer

Valid User for the Server (e.g. Stack Overflow, data sets)

Application Developer, Valid User for Data

Administrator

RECENT ATTACKS

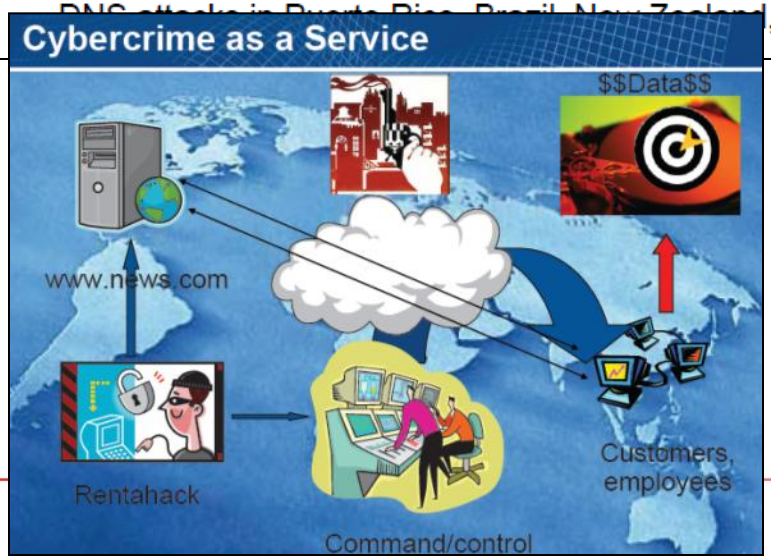
Higher Complexity

Source: IBM Silicon Valley Lab(2009)

protegrity

The Gartner 2010 CyberThreat Landscape

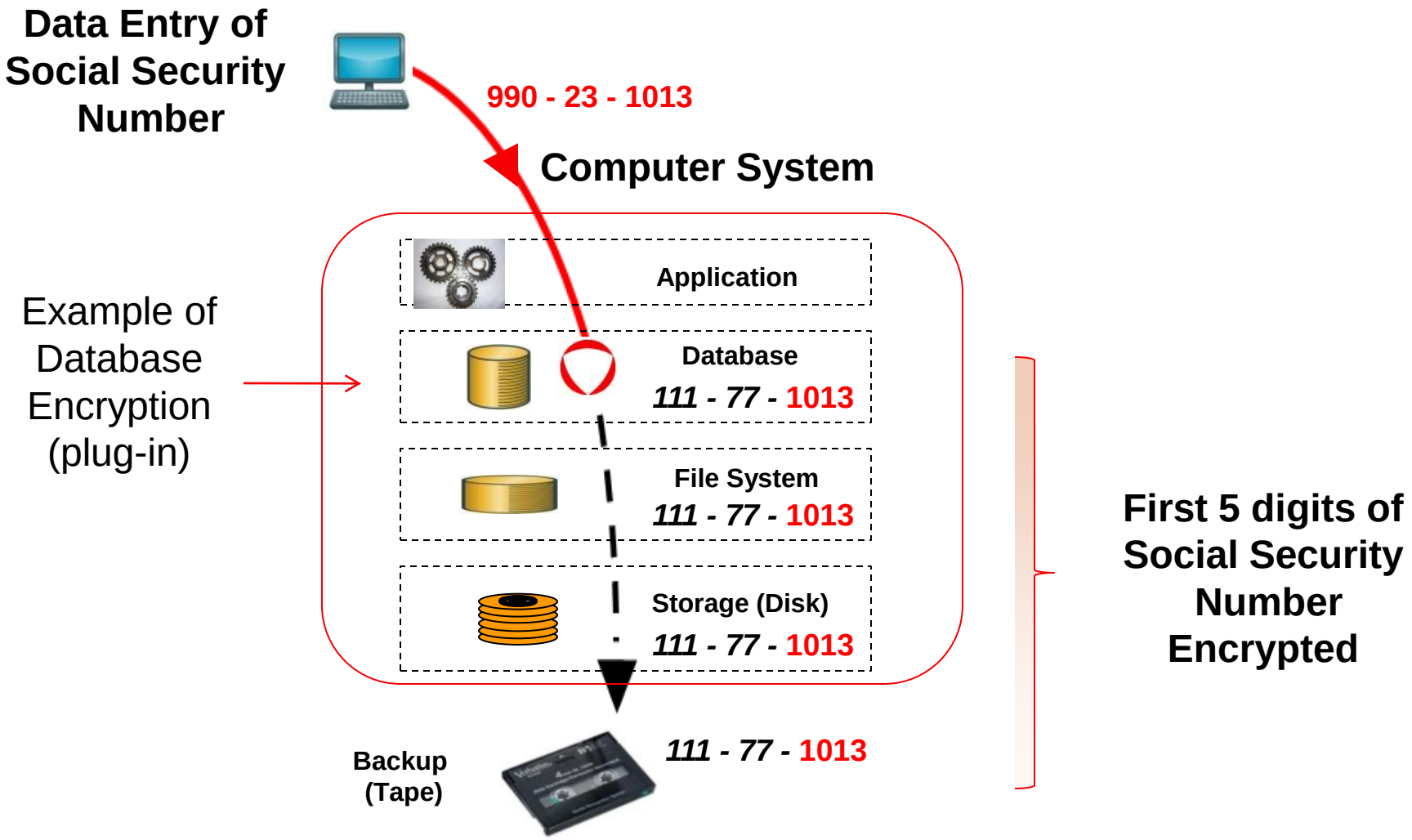
- Latest "largest ever" — Heartland Payment Systems
- DoD Bans USB after trojan on thumb drive
- Security incidents rise 24.7% at educational institutes
- Conficker hits medical machinery
- Worms hit Twitter, Facebook, LinkedIn, MySpace
- "Chinese attackers" steal jet fighter data
- DNS attacks in Puerto Rico, Brazil, New Zealand, U.S.



The Aha Slide

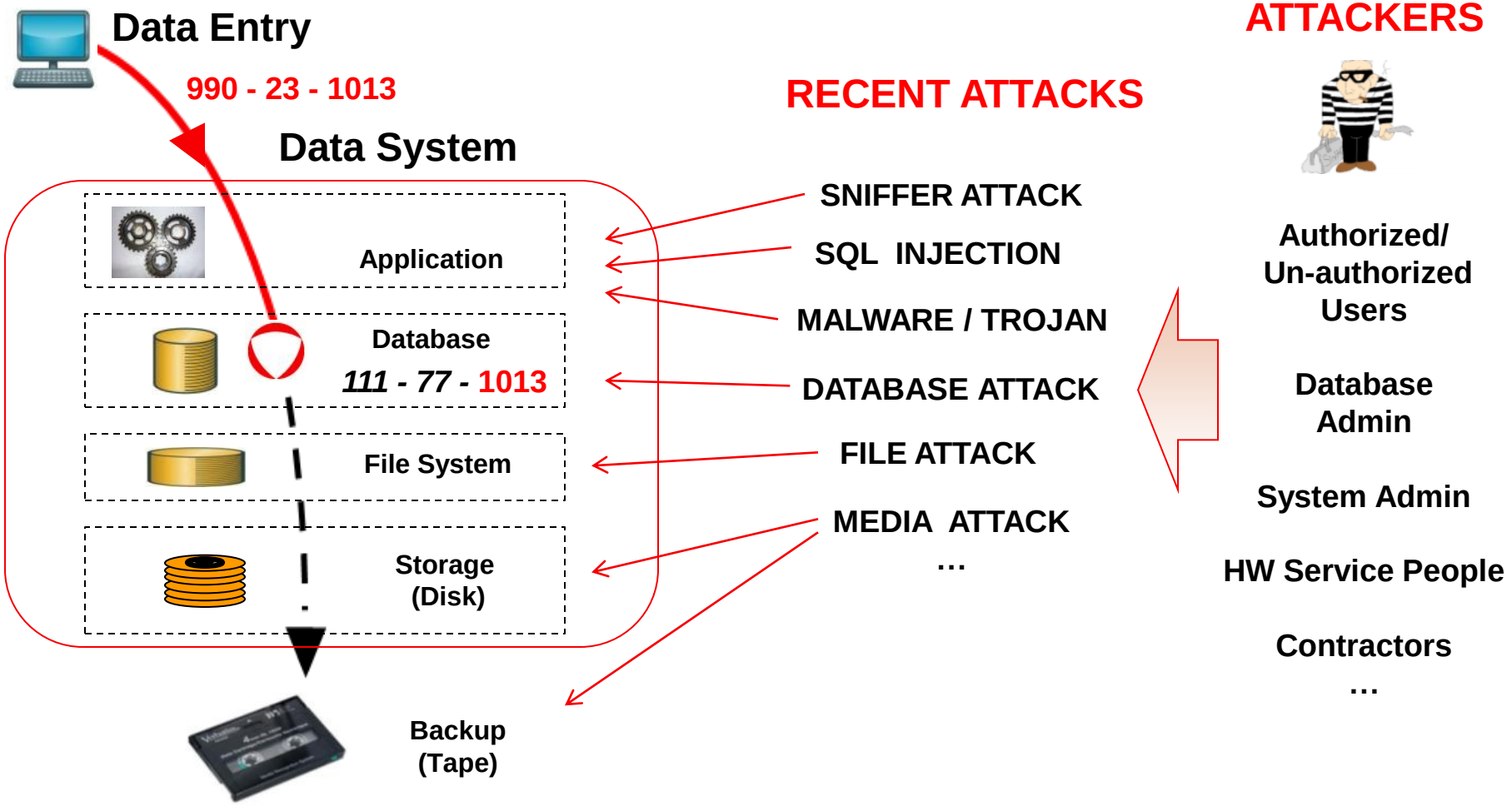
- We have met the threat and they are us.
  - New processes
  - New technologies
  - Complacency
- You need very different armor to survive a sniper's rifle shot than you do for a hailstorm.
- Threats will always change faster than user behavior

Step 4: Choose Your Defenses – An Example



Step 4: Choose Your Defenses

Where is data exposed to attacks?



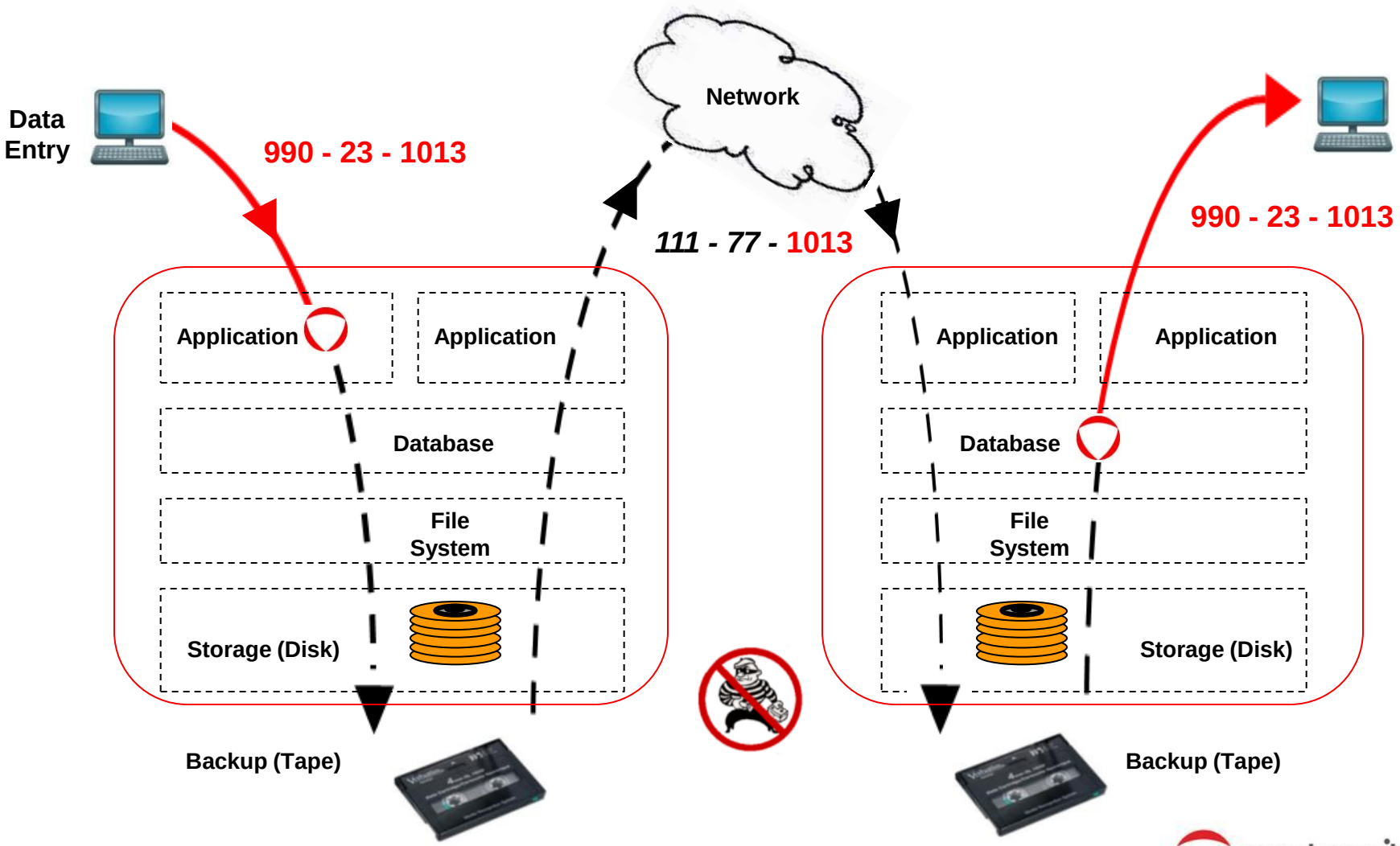
Unprotected sensitive information:

Protected sensitive information

protegrity

Step 4: Choose Your Defenses – Protect the Data Flow

Mitigation at the Right System Layer



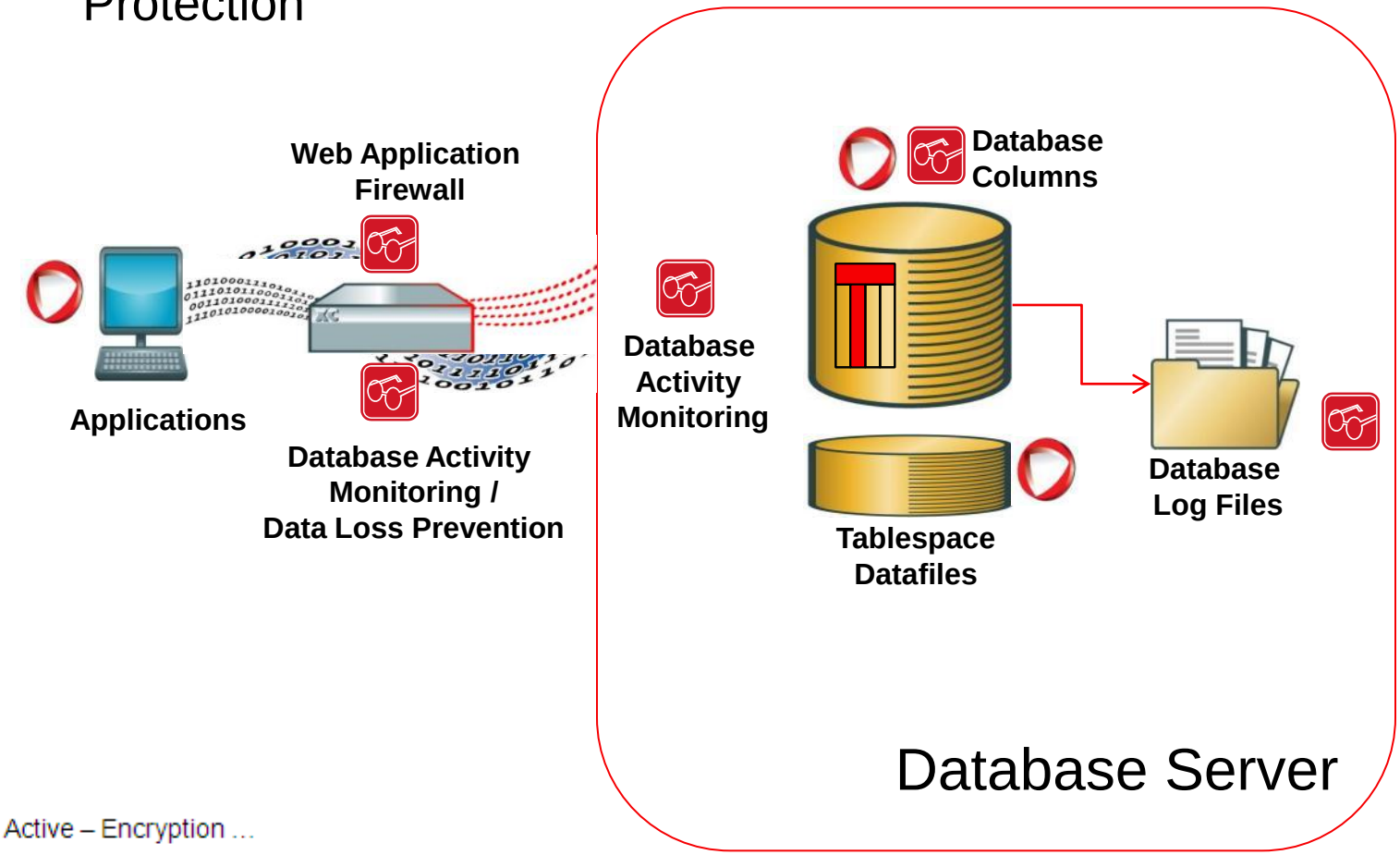
Unprotected sensitive information:

Protected sensitive information

protegrity

Protecting Data in the Enterprise Data Flow

Passive Approaches and Active Approaches = End-To-End Protection



Active – Encryption …

Passive – Monitoring …

protegrity

Passive Database Protection Approaches

Operational Impact Profile

Database Protection Approach	Performance	Storage	Security	Transparency	Separation of Duties
Web Application Firewall	●	●	◔	●	○
Data Loss Prevention	●	●	◔	●	○
Database Activity Monitoring	●	●	◔	●	○
Database Log Mining	◔	●	○	●	○

Best ● ◕ ◔ ◒ ◔ ◔ ○ Worst

protegrity

Active Database Protection Approaches

Operational Impact Profile

Database Protection Approach	Performance	Storage	Security	Transparency	Separation of Duties
Application Protection - API					
Column Level Encryption; FCE, AES, 3DES					
Column Level Replacement; Tokens					
Tablespace - Datafile Protection					

Best  Worst

protegrity

Step 4: Choose Your Defenses - New Protection Models

Format Controlling Encryption

Example of Encrypted format:

*111-22-**1013***

Application Databases

Key Manager

Data Tokenization

Example of Token format:

*1234 1234 1234 **4560***

Application Databases

Token

Token Server

Key Manager

protegrity

FCE Considerations

- Unproven level of security – makes significant alterations to the standard AES algorithm
- Encryption overhead – significant CPU consumption is required to execute the cipher
- Key management – is not able to attach a key ID, making key rotation more complex - SSN
- Some implementations only support certain data (based on data size, type, etc.)
- Support for “big iron” systems – is not portable across encodings (ASCII, EBCDIC)
- Transparency – some applications need full clear text

FCE Use Cases

- Suitable for lower risk data
- Compliance to NIST standard not needed
- Distributed environments
- Protection of the data flow
- Added performance overhead can be accepted
- Key rollover not needed – transient data
- Support available for data size, type, etc.
- Point to point protection if “big iron” mixed with Unix or Windows
- Possible to modify applications that need full clear text – or database plug-in available

Tokenization Considerations

- Transparency – not transparent to downstream systems that require the original data
- Performance & availability – imposes significant overhead from the initial tokenization operation and from subsequent lookups
- Performance & availability – imposes significant overhead if token server is remote or outsourced
- Security vulnerabilities of the tokens themselves – randomness and possibility of collisions
- Security vulnerabilities typical in in-house developed systems – exposing patterns and attack surfaces

Tokenization Use Cases

- Suitable for high risk data – payment card data
- When compliance to NIST standard needed
- Long life-cycle data
- Key rollover – easy to manage
- Centralized environments
- Suitable data size, type, etc.
- Support for “big iron” mixed with Unix or Windows
- Possible to modify the few applications that need full clear text – or database plug-in available

Data Protection in the Enterprise – Implementation Example

Need partial Information in clear:

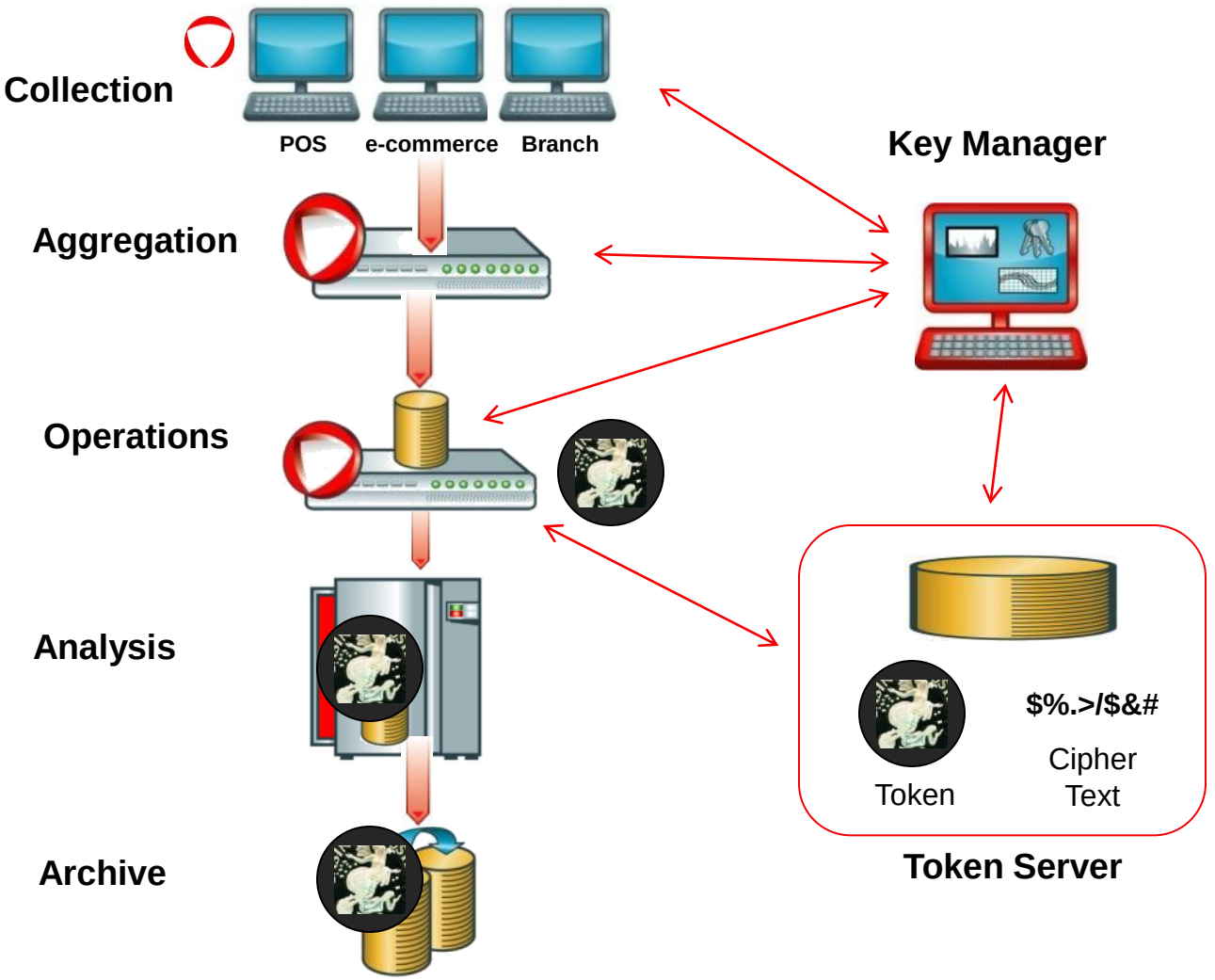
1234 1234 1234 **4560**

Need full Information in clear:

55 49 9437 0789 4560

Can use stored protected value:

1234 1234 1234 **4560**



Token Cipher

Step 4: Choose Your Defenses – Strengths/Weaknes

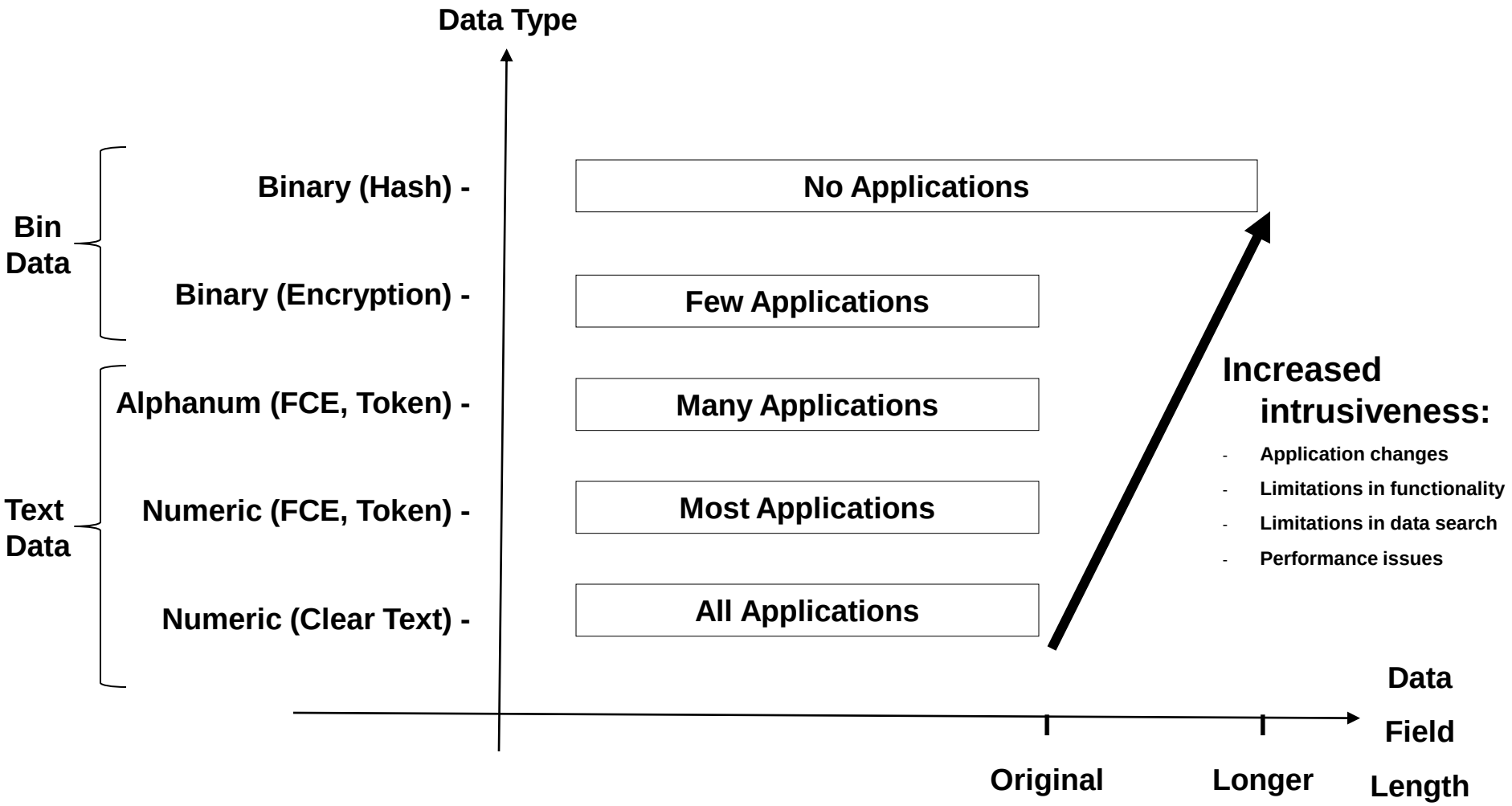
Data Protection Options	Performance	Storage	Security	Transparency
Clear	●	●	○	●
Monitoring + Blocking + Masking	◔	●	◔	◔
Format Controlling Encryption	◒	●	◒	◔
Strong Encryption *	◔	◔	●	◒
Tokens *	◔	●	●	◔
Hash *	◔	◔	●	○

Best ● ◔ ◒ ◔ ○ Worst

*: Compliant to PCI DSS 1.2 for making PAN unreadable

protegrity

Applications are Sensitive to the Data Format



This is a generalized example

protegrity

Enterprise View of Different Protection Options

Evaluation Criteria	Strong Encryption	Formatted Encryption	Token
Disconnected environments	●	●	○
Distributed environments	●	●	◔
Performance impact when loading data	●	◔	◔
Transparent to applications	●	◒	◒
Expanded storage size	◒	●	◔
Transparent to databases schema	◒	●	●
Long life-cycle data	◔	◔	●
Unix or Windows mixed with “big iron” (EBCDIC)	●	◔	●
Easy re-keying of data in a data flow	◒	◔	●
High risk data	●	○	●
Security - compliance to PCI, NIST	●	○	●

Best ● ◔ ◒ ◔ ○ Worst

protegrity

Vendors/Products Providing Database Protection

Feature	3rd Party	Oracle 9	Oracle 10	Oracle 11	IBM DB2	MS SQL
Database file encryption	●	○	○	●	●	◒
Database column encryption	●	●	●	●	●	◒
Column encryption adds 32-52 bytes (10.2.0.4, 11.1.0.7)	●	○	○	○	○	◒
Formatted encryption	●	○	○	○	○	○
Data tokenization	●	○	○	○	○	○
Database activity monitoring	◔	○	○	○	●	○
Multi vendor encryption	●	○	○	○	○	○
Data masking	●	○	◒	●	●	○
Central key management	●	○	○	○	●	◒
HSM support (11.1.0.7)	●	○	○	●	◒	◒
Re-key support (tablespace)	●	○	○	○	○	○

Best ● ◔ ◒ ◔ ○ Worst

protegrity

Column Encryption Solutions – Some Considerations

Area of Evaluation	3rd Party	Oracle 10 TDE	Oracle 11 TDE
Performance, manage UDT or views/triggers	◔	●	●
Support for both encryption and replication	●	○	○
Support for Oracle Domain Index for fast search	◒	○	○
Keys are local; re-encryption if moving A -> B	●	○	○
Separation of duties/key control vector	●	○	○
Encryption format specified	●	○	○
Data type support	●	◒	◒
Index support beyond equality comparison	●	○	○
HSM (hardware crypto) support (11.1.0.6)	●	○	●
HSM password not stored in file	●	○	○
Automated and secure master key backup procedure	●	○	○
Keys exportable	●	○	○

Best ● ◔ ◒ ◕ ○ Worst

protegrity

Oracle Master Key Management

EXTERNAL SECURITY MODULE SUPPORT BY DATABASE VERSION			
DATABASE VERSION	**MASTER KEY FOR ...**	**... IN ORACLE WALLET**	**... IN HSM**
Oracle Database 10gR2	Column Encryption	Yes	No
Oracle Database 11gR1 (11.1.0.6)	Column Encryption	Yes	Yes
	Tablespace Encryption	Yes	No
Oracle Database 11gR1 (11.1.0.7)	Column Encryption	Yes	Yes
	Tablespace Encryption	Yes	Yes (no re-key)

RE-KEY SUPPORT				
	TDE COLUMN ENCRYPTION		**TDE TABLESPACE ENCRYPTION**	
	MASTER KEY	**TABLE KEYS**	**MASTER KEY**	**TABLESPACE KEYS**
Re-key support	Yes	Yes	No	No

protegrity

TDE Tablespace or TDE Column Encryption?

CHOOSE TDE COLUMN ENCRYPTION IF ...:	CHOOSE TDE TABLESPACE ENCRYPTION IF ...:
Keys need to be rotated on a semi frequent basis	Key rotation is not required
Location of sensitive information is known	Location of sensitive information is unknown
Less than 5% of all application columns are encryption candidates.	Most of the application data is deemed sensitive, or **multiple** national and international security and privacy mandates apply to your industry
Data type and length is supported by TDE column encryption	Not all data types that hold sensitive information are supported by TDE column encryption
Encryption candidates are not foreign-key columns	Encryption candidates are foreign key columns
Indexes over encryption candidates are normal B-tree indexes	Indexes of encryption candidates are functional indexes
Application does not perform range scans over encrypted data	Application searches for ranges of sensitive data
Increase in storage by 1 to 52 bytes per encrypted value	No storage increase acceptable
Performance impact depends on percentage of encrypted columns; how often the encrypted values are selected or updated, the size of encrypted data, and other variables.	Constant performance impact below 10%

Vendors Providing Strong Encryption

Feature	Vendor A	Vendor B	Vendor C	Oracle	Vendor D	Vendor E
Software solution	●	●	●	●	○	○
HSM support	●	●	○	●	●	●
Database support	●	◔	●	◔	◒	◔
File encryption support	●	◔	●	○	◔	◕
Performance	●	◒	◕	●	◔	●
FIPS	◒	◒	○	◒	●	●
Availability	●	◒	●	●	○	◒
Central key management	●	●	●	○	●	●

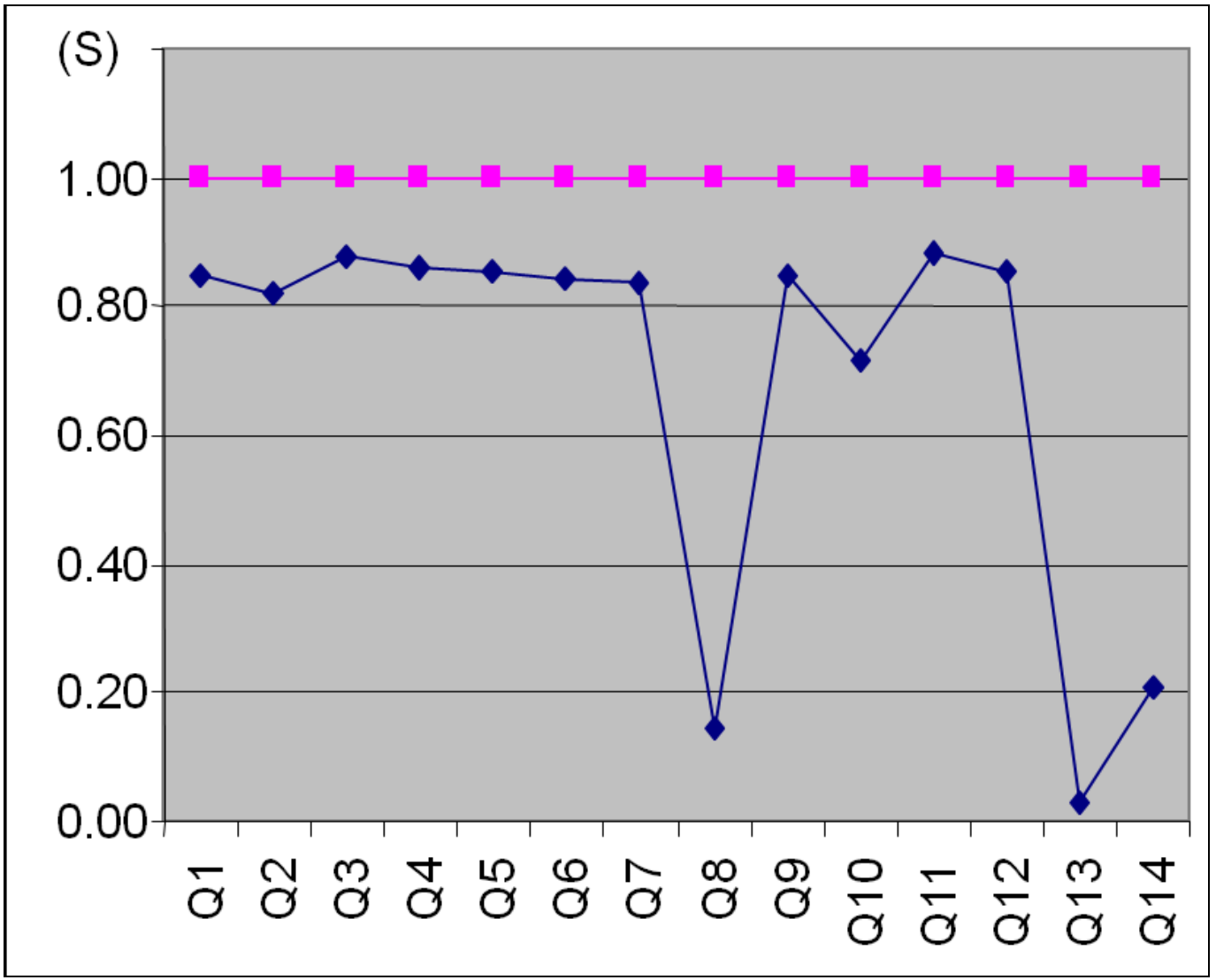
Best ● ◕ ◒ ◔ ○ Worst

protegrity

How to encrypt indexed columns in Oracle

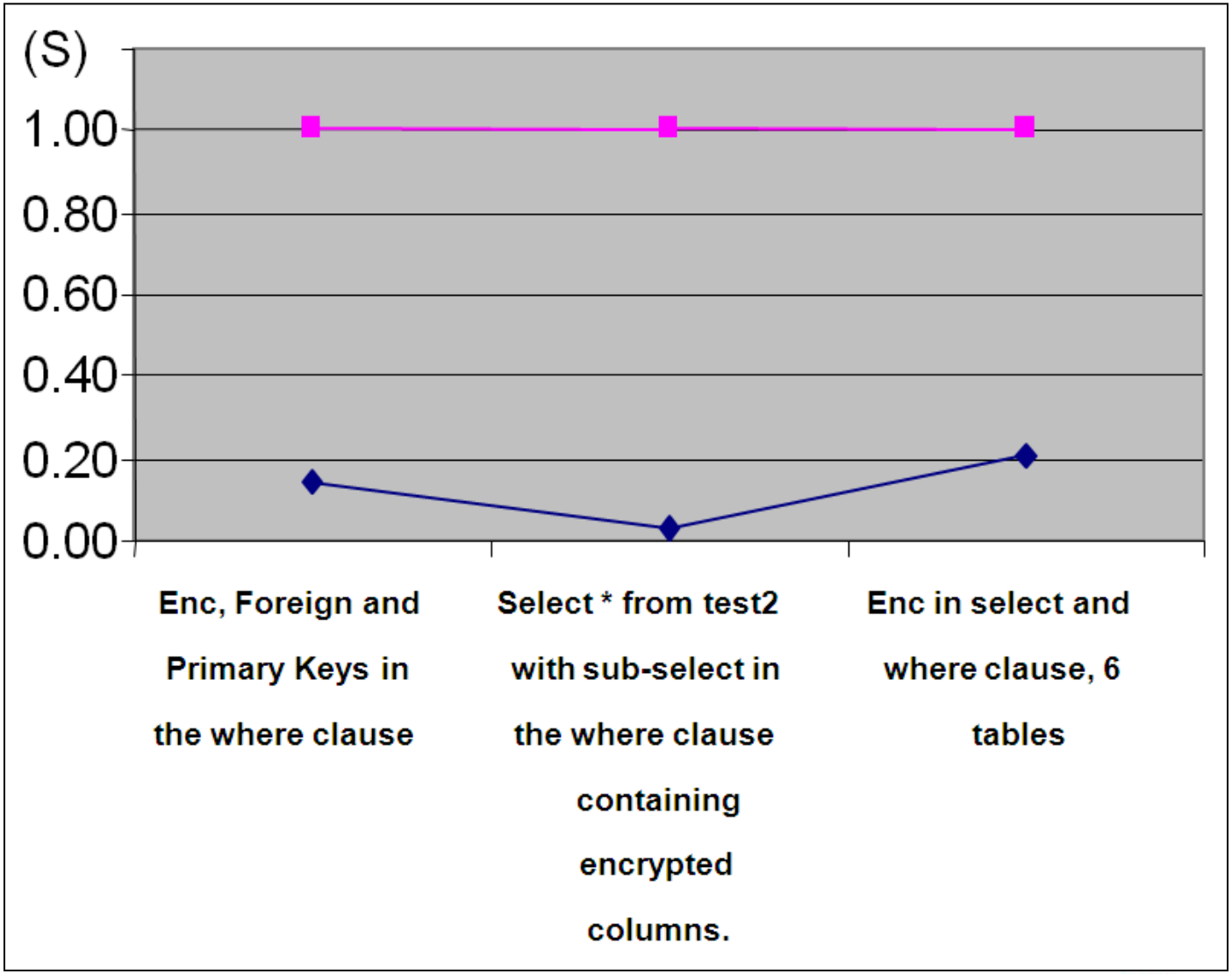
- Some advanced solutions enable encrypted data to be searched without the overhead of first decrypting into clear text.
- Only the result set is decrypted to clear text.
- Most vendor solutions force the data to be decrypted before being searched.
- An index search for an exact match of an encrypted value within a column is possible with several vendor solutions, provided that the same initialization vector is used for the entire column.
- On the other hand, searching for partial matches on encrypted data within a database can be challenging and can result in full table scans if support for accelerated index-search on encrypted data is not used.

Oracle Domain Index



protegrity

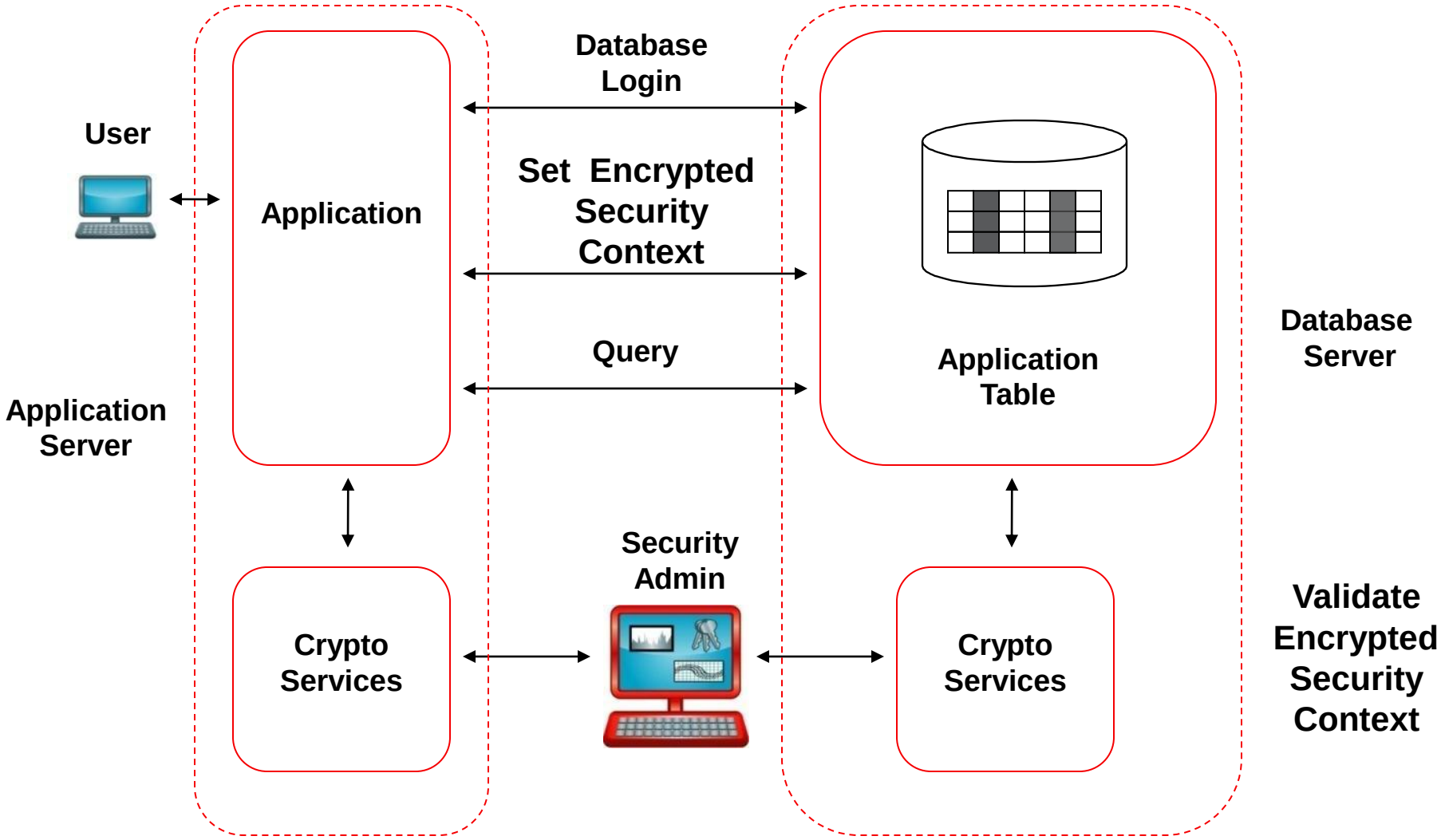
Oracle Domain Index



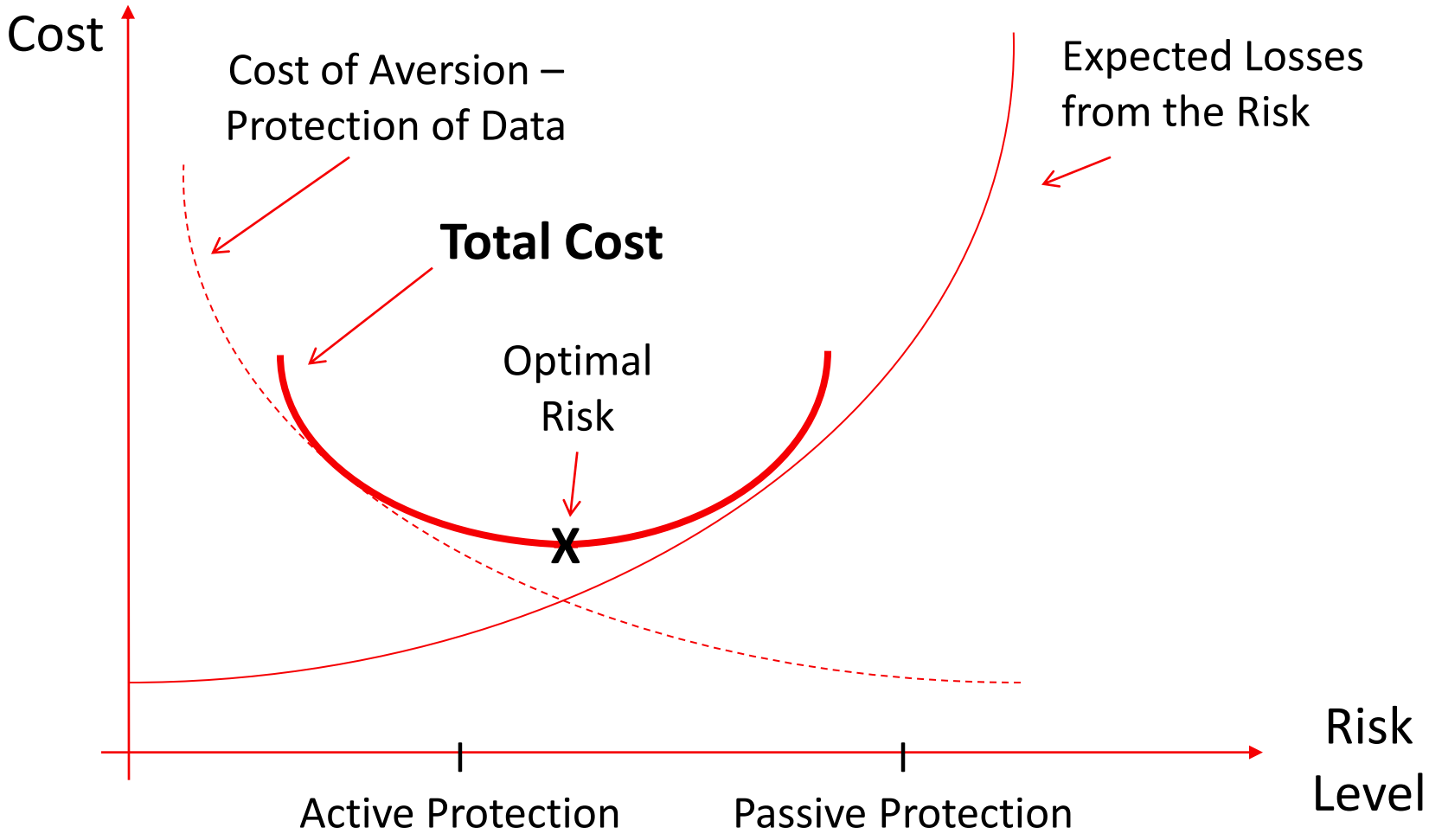
A More Secure Login – US Government Example

- Extended access checks needed
  - V$SESSION parameters may not be secure
  - Not based on properly authenticated data
  - Only data collected during the authentication process
- Client can manipulate parameters
  - PL/SQL - module name …
  - Some may be set by the client process
    - If calling with JDBC, it's possible to set many of the V$SESSION parameters
  - Simply renaming a process may fool the client DLL that the application is something else than what is seen
  - If sqlplus.exe is not allowed, rename it to something that is allowed
  - If sqlplus.exe is allowed, rename the false program into this name
  - The program name is only VARCHAR2(48), so the path is not included.

Secure Login – US Government Example



Step 4: Choose Your Defenses – Find the Balance

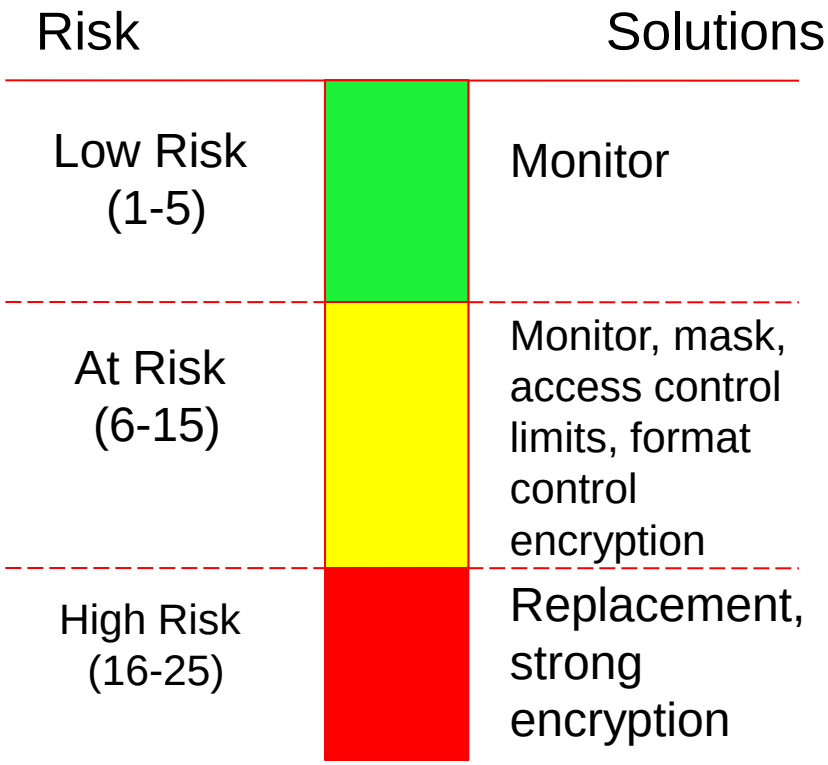


protegrity

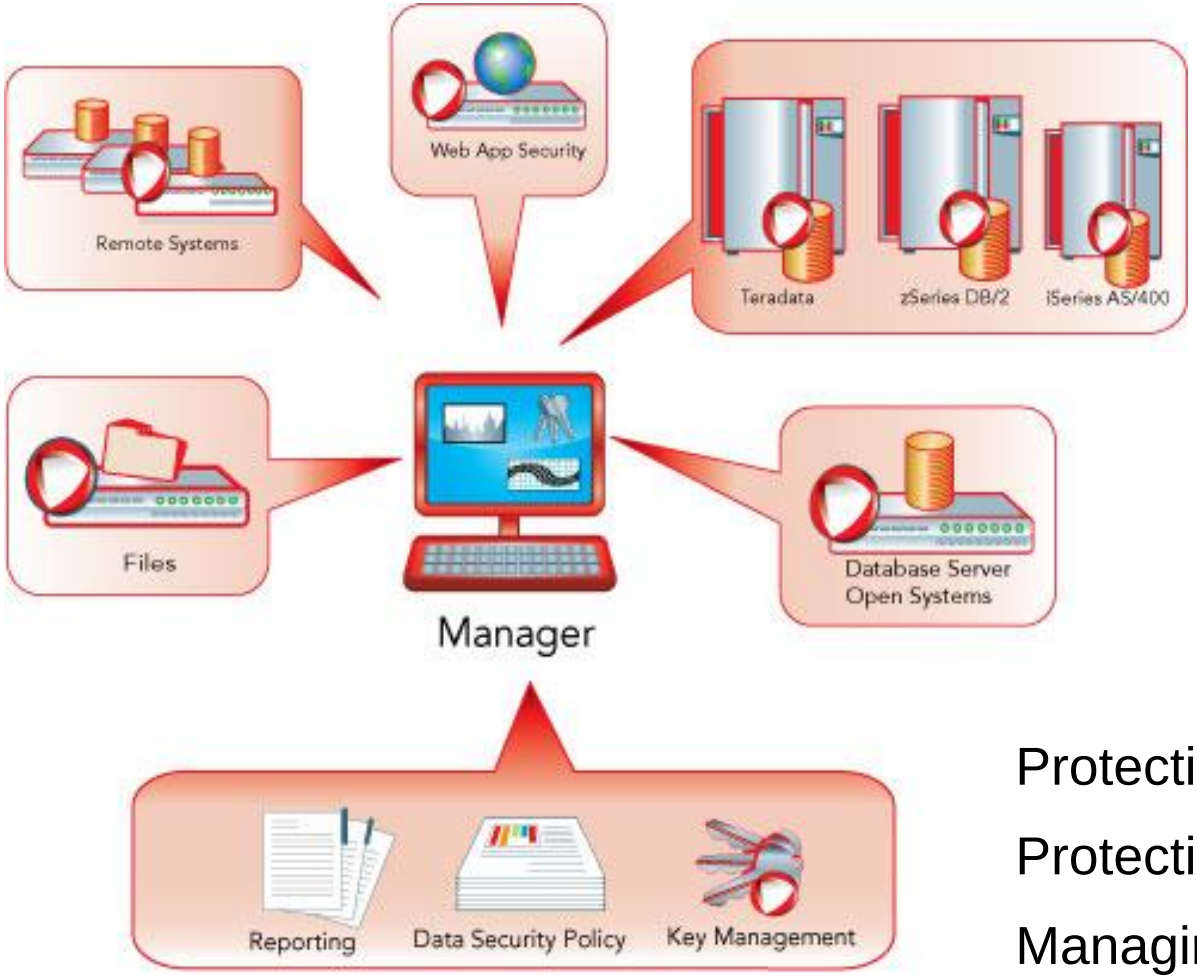
Matching Data Protection Solutions with Risk Level

Data Field	Risk Level
Credit Card Number	25
Social Security Number	20
CVV	20
Customer Name	12
Secret Formula	10
Employee Name	9
Employee Health Record	6
Zip Code	3

Select risk-adjusted solutions for costing



Step 5: Deployment – A Staged Security Rollout



Protecting data

Protecting web applications

Managing data security

protegrity

Step 6: Crunch the Numbers – Conclusion

- Risk-adjusted data security plans are cost effective
- Switching focus to a holistic view rather than security silo methodology
- Understanding of where data resides usually results in a project to reduce the number of places where sensitive data is stored
- Protect the remaining sensitive data with a comprehensive data protection solution

Table Space / File Encryption

Oracle® Database Advanced Security Administrator's Guide 11*g* Release 1 (11.1)

All data in an encrypted tablespace is stored in encrypted format on the disk. Data is transparently decrypted for an authorized user having the necessary privileges to view or modify the data. A database user does not need to know if the data in a particular table is encrypted on the disk. In the event that the data files on a disk or backup media gets stolen, the data is not compromised.

Tablespace encryption uses the transparent data encryption architecture to transparently encrypt (and decrypt) tablespaces. The tablespace encryption master key is stored in the same Oracle wallet that is used to store the transparent data encryption master key. This tablespace encryption master key is used to encrypt the tablespace encryption key, which in turn is used to encrypt and decrypt data in the tablespace.

The encrypted data is protected during operations like `JOIN` and `SORT`. This means that the data is safe when it is moved to temporary tablespaces. Data in undo and redo logs is also protected.

Tablespace encryption also allows index range scans on data in encrypted tablespaces. This is not possible with column-based transparent data encryption.

Encrypts IBM Databases. Sort Of.

Securosis

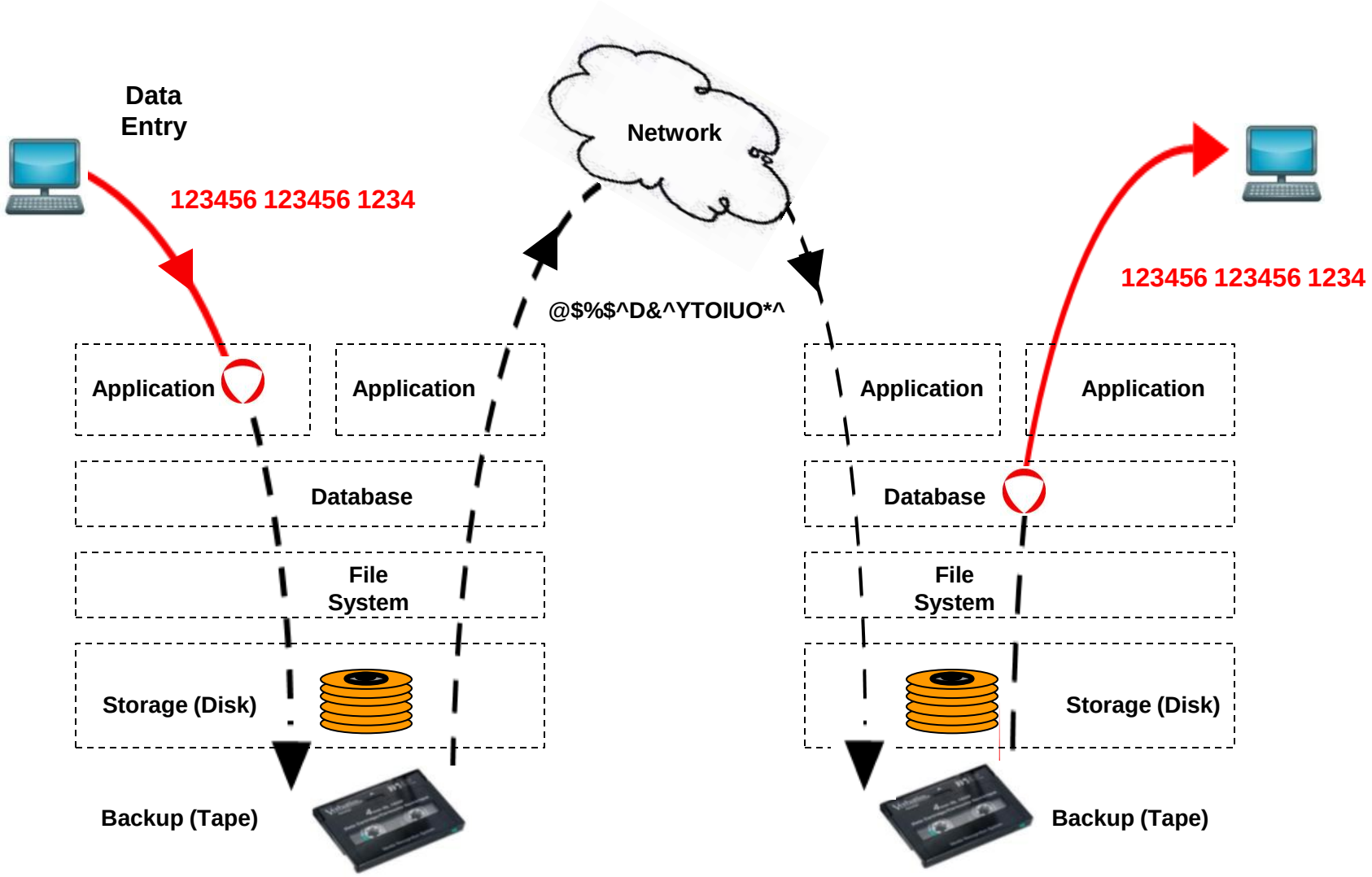
Remember, I think encrypting the database files, especially when used with Database Activity Monitoring, is an extremely effective security control. But it doesn't replace field level encryption, not in the long run.

The role of file level encryption for databases is media protection first, with a little separation of duties second. It protects that database on disk and in backups. It also limits who can access the raw database files, but offers no protection for authorized users and administrators in the database.

The role of field (column) level encryption is to provide separation of duties within the database. You can protect sensitive fields from those who have database access, including protection against database administrators.

Two kinds of encryption. Two different roles. Two different problems solved.

Data Protection Implementation - Enforcement Points



Unprotected sensitive information:

Protected sensitive information

protegrity

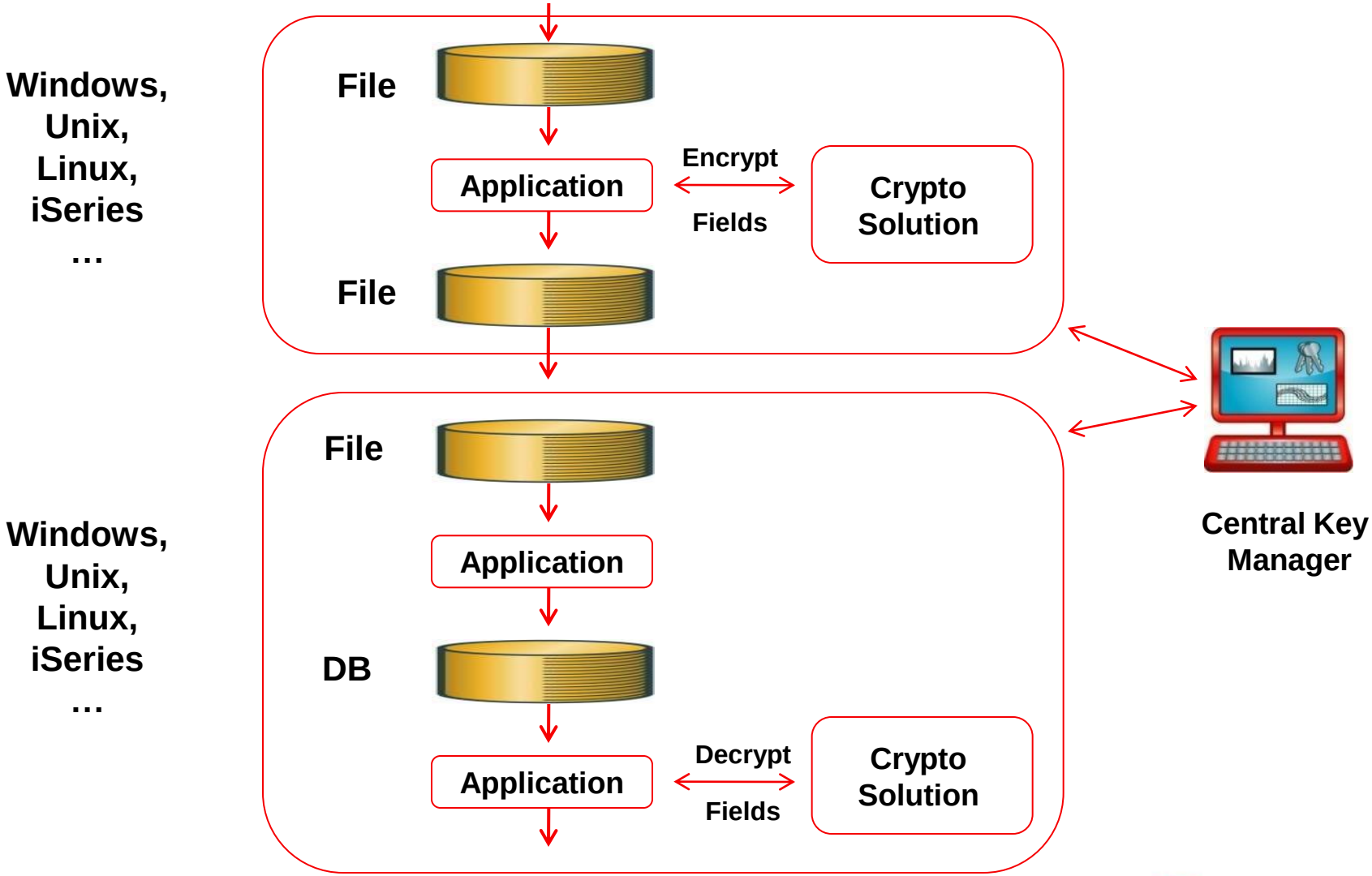
Data Protection Implementation – System Layers

System Layer	Performance	Transparency	Security
Application	●	○	●
Database	◒	◔	◔
File System	◔	●	◒

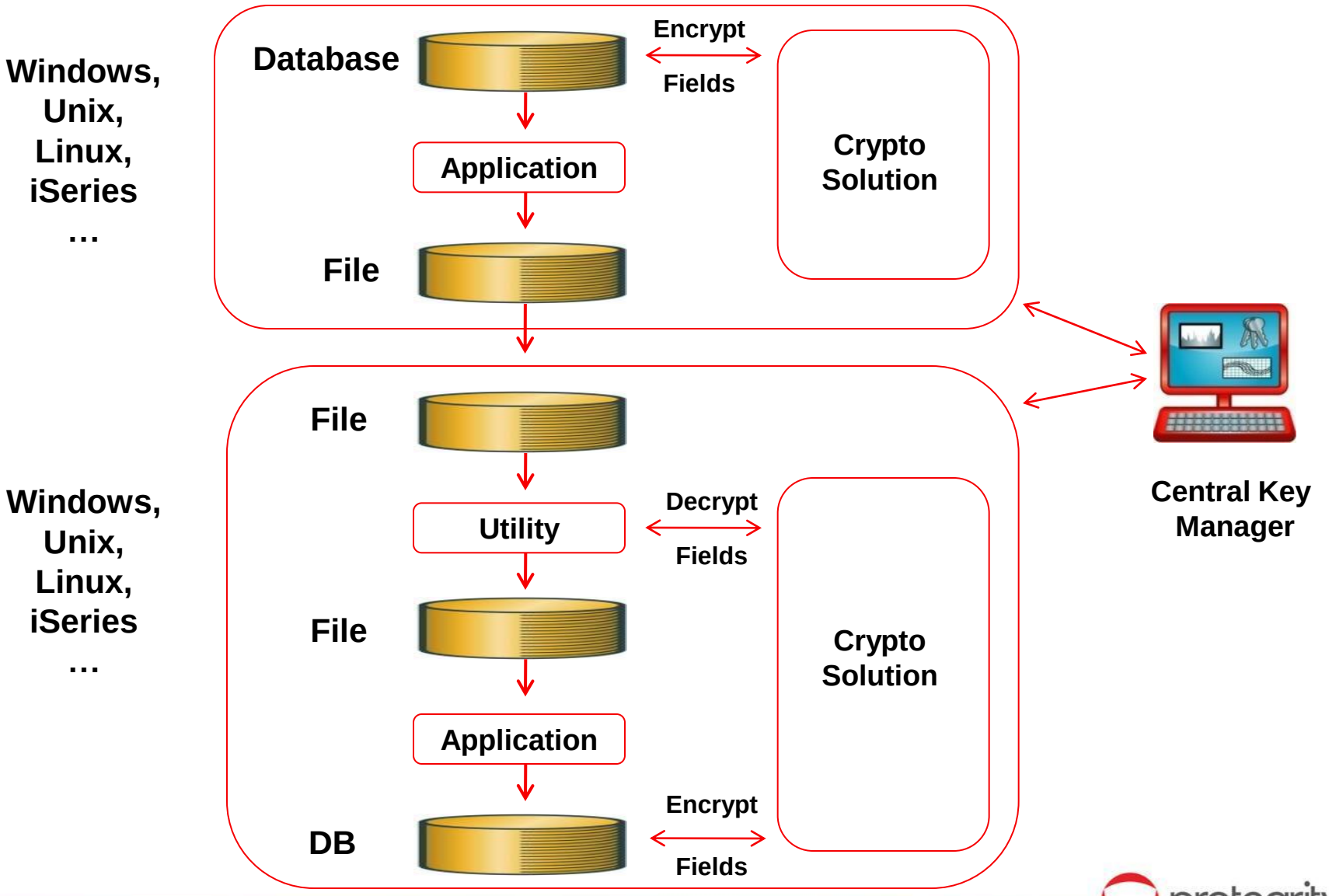
Topology	Performance	Scalability	Security
Local Service	●	●	◔
Remote Service	○	○	●

Best ● ◔ ◒ ◔ ○ Worst

Field Encryption – Protecting the Data Flow

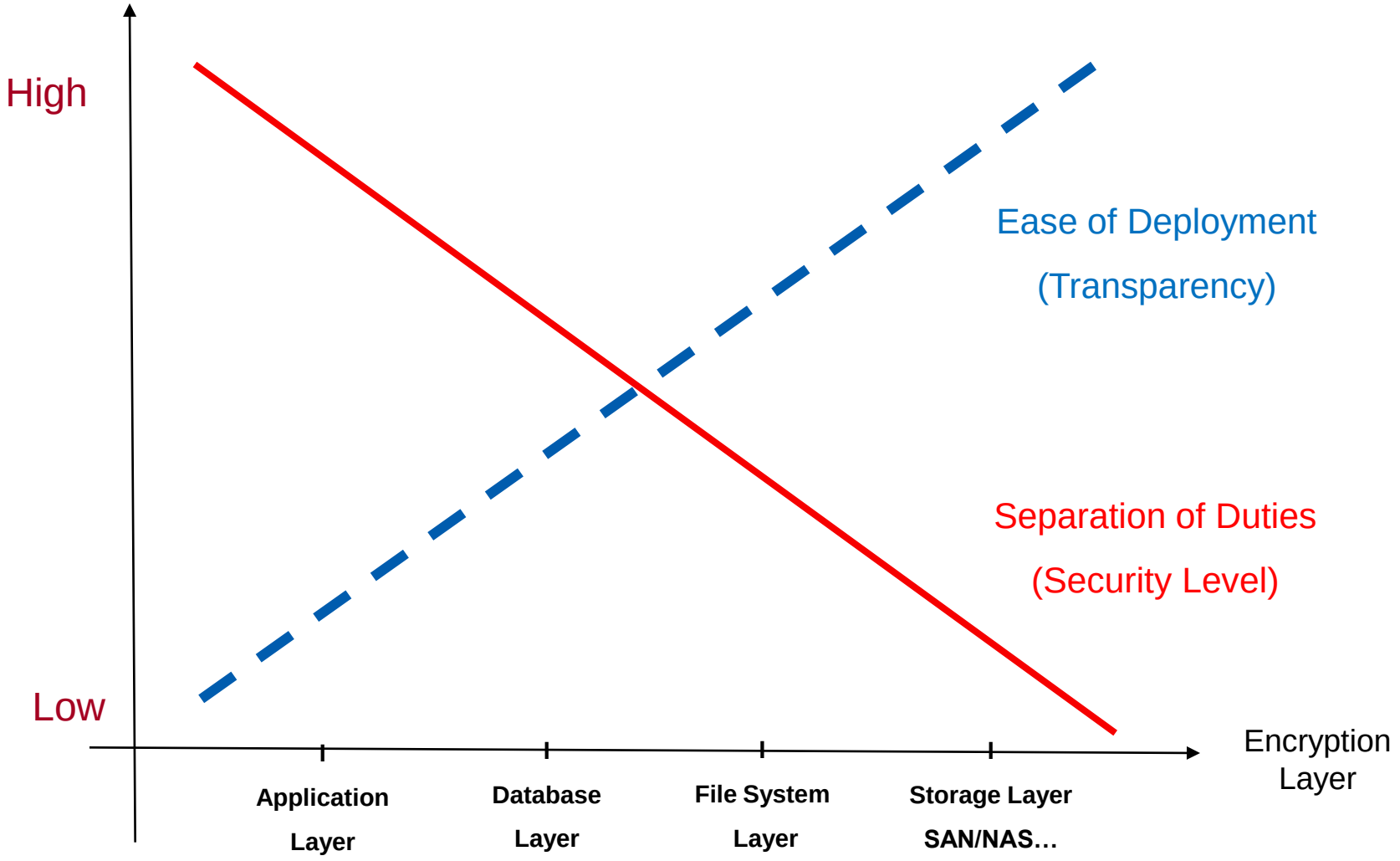


Transparent Encryption – No Application Changes



protegrity

Generalization: Encryption at Different System Layers



protegrity

Questions?

If you would like a copy of the slides,
please email
ulf.mattsson@protegrity.com

protegrity

protecting your data.
protecting your business.

Appendix

protegrity

protecting your data.
protecting your business.

States move to mandate encryption of sensitive personal data

Posted on | *March 2, 2009* | *add a comment*

First came data loss disclosure requirements and credit freeze laws. Now comes data encryption laws. In response to the continuing wave of data heists, Massachusettes and Nevada are leading the way in passing new state laws dictating what businesses must do to protect credit card transaction records and other personal data. These new laws go much further than the hard-won laws in more than 30 states requiring

Data Encryption Management New York NY

Encryption is becoming a part of the corporate landscape, partly out of necessity and partly because state laws are forcing it upon companies. But laws differ by state and, at this stage in the game, companies cannot assume that just because they have encrypted data or implemented encryption key management that they are either completely

Oracle RMAN Encryption and Compression

	BACKUP WITH COMPRESSION	BACKUP WITH ENCRYPTION	BACKUP WITH COMPRESSION AND ENCRYPTION
Data not encrypted	Data compressed	Data encrypted	Data compressed first, then encrypted
Data encrypted with TDE column encryption	Data compressed; encrypted columns are treated as if they were not encrypted.	Data encrypted; double encryption of encrypted columns.	Data compressed first, then encrypted; encrypted columns are treated as if they were not encrypted; double encryption of encrypted columns
Data encrypted with TDE tablespace encryption	Encrypted blocks are decrypted, compressed, and re-encrypted; Clear text blocks are compressed and encrypted (after compression, they cannot be distinguished from encrypted data).	Encrypted blocks are passed through to the backup unchanged; clear text blocks are encrypted for backup.	Encrypted blocks are decrypted, compressed, and re-encrypted; Clear text blocks are compressed and encrypted.

Oracle Domain Index

Query	Test Description	Number rows returned	ENC with DI (secs)	ENC w/o DI (secs)	Comments	Total decrypts (est.)	Avg cost per decrypt (msecs)
Q1	Straight select from sample2 table. 3 encrypted columns in the select.	10069	6.000	7.08	Overhead due to decryption of 3 fields * the number of rows returned.Explain plans identical.	30207	0.04
Q2	Straight select from test2 table. 3 encrypted columns in the select.	21999	11.500	14.08	Overhead due to decryption of 3 fields * the number of rows returned.Explain plans identical.	65997	0.04
Q3	Straight select from result2 table. 4 encrypted columns in the select.	19998	14.070	16.09	Overhead due to decryption of 4 fields * the number of rows returned.Explain plans identical.	79992	0.03
Q4	Join of Sample2, Test2 and Results2 Table. No encrypted columns in where clause. 5 encryped columns in the select clause.	5037	6.010	7.00	Overhead due to decryption of 5 fields * the number of rows returned.Explain plans similar.	25185	0.04
Q5	Select 3 encrypted columns from sample2 table. No joins, 1 encrypted column, S62_ENC, in the where clause (=).	400	2.600	3.05	Overhead includes full table access and decrypting s62_enc * 10770 rows even though only 400 rows are returned.	11570	0.04

Oracle Domain Index

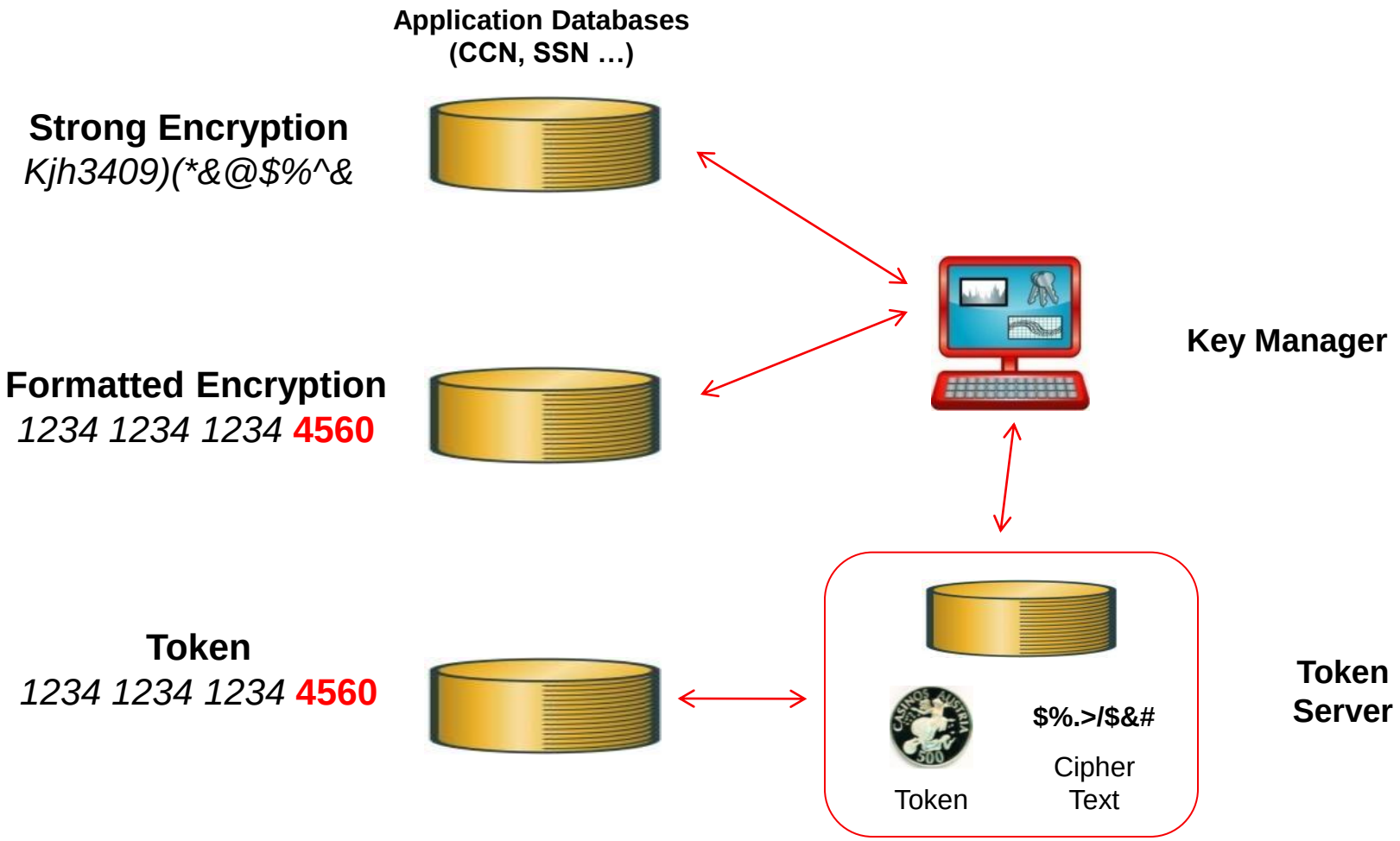
Query	Test Description	Number rows returned	ENC with DI (secs)	ENC w/o DI (secs)	Comments	Total decrypts (est.)	Avg cost per decrypt (msecs)
Q6	Select from test2 table. No joins, 1 encrypted column (T2_ENC) in the where clause (< > range	3090	5.900	7.02	Overhead includes decrypting t2_enc * 22,000 rows even though only 3090 rows are returned. T10_enc was decrypted for each returned row.	25090	0.04
Q7	Select from sample2 table. No joins, 2 encrypted column (S62_ENC = , S26_ENC >) in the where clause.	1000	5.070	6.07	Overhead includes a full table scan and decrypting s62_enc for every row in the table (25000 times). S35_enc and s26_enc were decrypted for each row returned.	27000	0.04
Q8	Select from sample2 & boted2 table. 2 encrypted column in the select clause. Encrypted columns, Foreign and Primary Keys in the where clause (sample2.S62_ENC = boted2.B_PK1_ENC).	899	0.010	0.07	Overhead includes decryption of 2 fields for every row returned (2 fields in sample2 and 1 field in boted2 during the join). Explain plans are identical.	2697	0.02
Q9	Select from test2 & analysis2 table. 1 encrypted column in the select clause. Encrypted columns, Primary and Foreign Keys in the where clause (test2.T2_ENC = analysis2.A_PK1_ENC) .	620	8.500	10.03	Overhead includes decryption of t2_enc field for every row in test2 and decryption of a_pk1_enc for every row in analysis2. Explain plan includes full table scan of analysis2 VS index range scan.	44913	0.03

Oracle Domain Index

Query	Test Description	Number rows returned	ENC with DI (secs)	ENC w/o DI (secs)	Comments	Total decrypts (est.)	Avg cost per decrypt (msecs)
Q10	Select from result2 & component2 table.5 encrypted column in the select clause. Encrypted columns, Foreign and Primary Keys in the where clause (result2.R_PK2_ENC = componet2.C_PK1_ENC).	920	10.000	14.01	Overhead includes decryption of r_pk2_enc field for every row in result2 and decryption of c_pk1_enc for every row in result2. Explain plans are similar.	68105	0.06
Q11	Select from Sample2 and Boted2 table. Outer join on encryted columns. (sample2.S62_ENC = boted2.B_PK1_ENC(+))	5599	2.700	3.06	Overhead includes decryption of s62_enc and b_pk1_enc field for every row returned due since the join is on the encrypted column. Explain plans are similar.	11198	0.03
Q12	Select from result2 table. 5 encrypted columns in the select clause. 5 Encrypted columns in the where clause.	600	12.000	14.01	Overhead includes decryption of s62_enc and b_pk1_enc field for every row returned due since the join is on the encrypted column. Explain plans are similar.	64065	0.03
Q13	Select * from test2 with sub-select in the where clause containing encrypted columns.	1300	0.030	1.05	Overhead includes decryption of 3 fields in test2 for every row returned and decryption of 1 field in for each row in analysis2 to satisfy the subquery. Explain plans are similar.	6034	0.17
Q14	Encrypted columns in select and where clause, 6 tables: Sample2, boted2, Test2, Analysis2, Results2 & Components2 joined in the from clause.	1197	1.050	5.04	Overhead includes decryption of seven fields for every row returned. Additional overhead decrypting fields during table joins but exact number of decryptions not known. Explain plans are similar.	18379	0.22

Data Protection Options
-
Use Cases

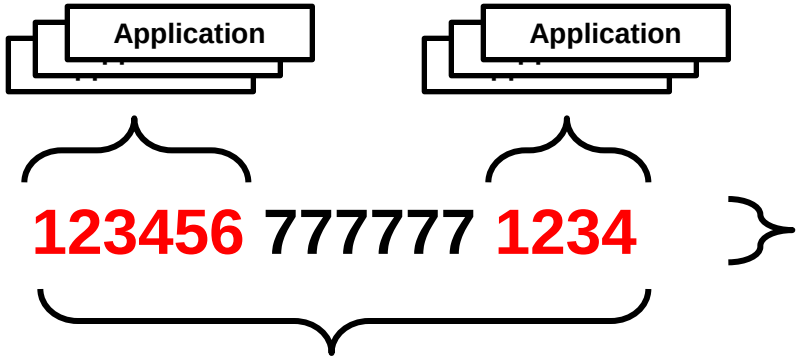
Data Protection Options in the Enterprise



Partial Encryption/Tokenizing - Example

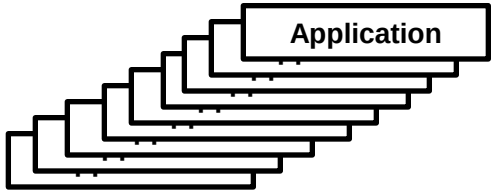
Some applications

- Partial clear data



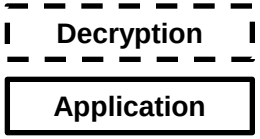
Many applications/tools

- Moving data around



Few applications

- Full clear data



protegrity

Data Protection Options – 3 Use Cases

Can use stored protected value:

1234 1234 1234 **4560**
Or
Kjh3409)(&@$%^&*

Application 1

Need partial Information in clear:

1234 1234 1234 **4560**

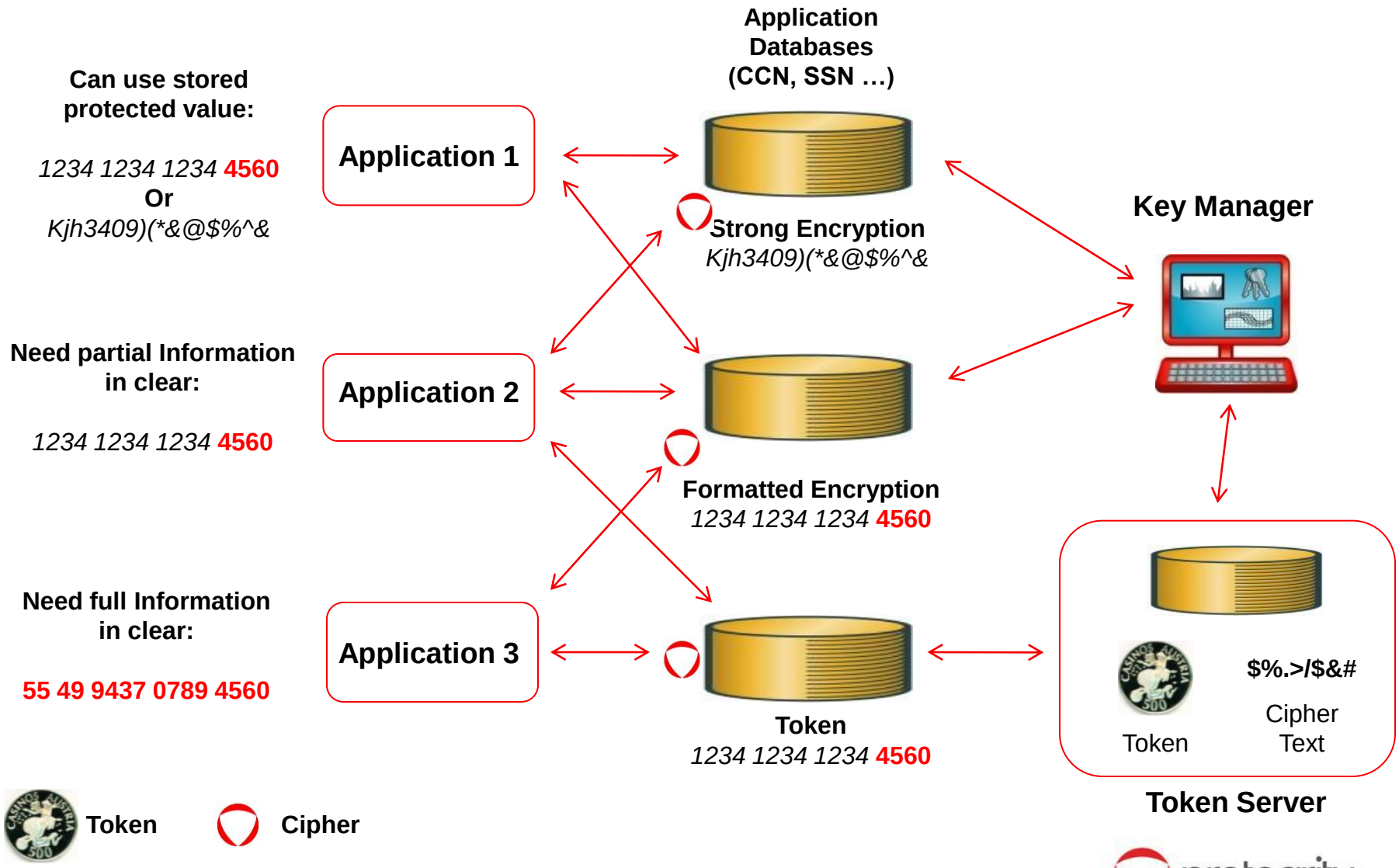
Application 2

Need full Information in clear:

55 49 9437 0789 4560

Application 3

How will different Protection Options Impact Applications?



Application Impact with Different Protection Options

Transparency

Type of Application	Strong Encryption	Formatted Encryption	Token
Can operate on the stored protected value (few)	●	●	●
Need partial information in clear (many)	○	●	●
Need full clear text information (few)	●	○	○

Security

Type of Application	Strong Encryption	Formatted Encryption	Token
Can operate on the stored protected value (few)	●	◒	●
Need partial information in clear (many)	◔	◒	●
Need full clear text information (few)	◔	◔	◔

Application Impact with Different Protection Options

Performance and scalability

Type of Application	Strong Encryption	Formatted Encryption	Token
Can operate on the stored protected value (few)	●	●	●
Need partial information in clear (many)	◔	●	●
Need full clear text information (few)	◔	◒	○

Availability

Type of Application	Strong Encryption	Formatted Encryption	Token
Can operate on the stored protected value (few)	●	●	●
Need partial information in clear (many)	◔	●	●
Need full clear text information (few)	◔	◔	○

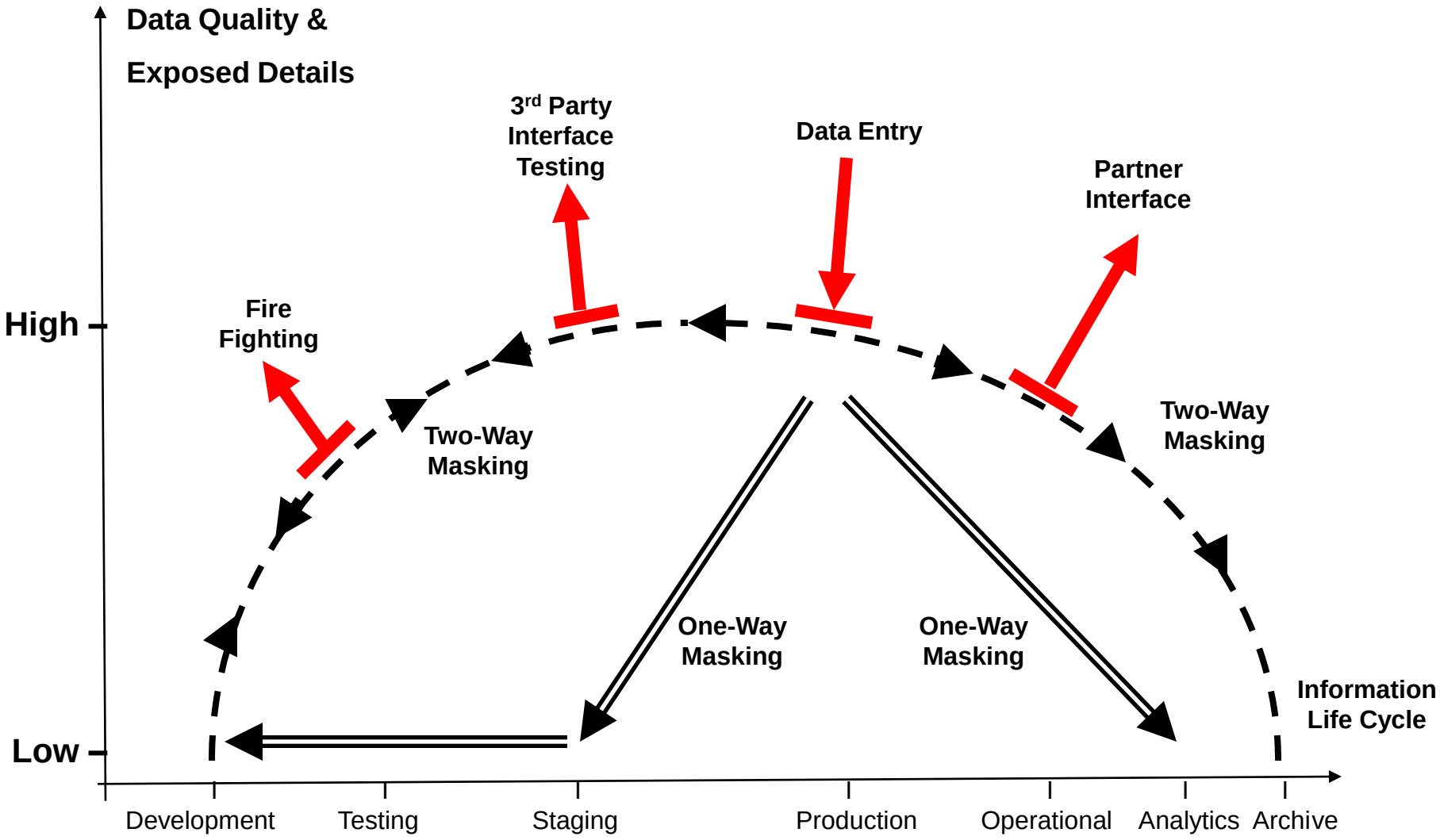
Case Studies

- **One of the most widely recognized credit and debit card brands in the world**
  - Their volume of data is in the multiple billions of rows and needed a solution that would not degrade performance.
- **Major financial institution**
  - Protecting high-worth clients financial information.
  - Central key management and separation of duties were of the utmost importance.
- **One of the world largest retailers**
  - Protecting the flow of sensitive credit card information from the store, through to back office systems and into the data warehouse and storage.
  - The central key management and ability to support thousands of stores was critical for this success.
  - Transparent to exiting applications.
  - Protect sensitive information in their data warehouse, operational systems and to files that reside across different platforms.

PCI Case Study – Large Retailer

- Minimal impact to the legacy environment
  - Encrypting PAN in the POS application and decrypting in HQ server
  - Encrypting PAN in databases, transparent to applications
  - Software encryption – 10 million transactions per second
- End-to-end encryption within the control of a single enterprise
  - Minimize modifications of applications, files and databases
  - Definition of “Strong cryptography” - PCI DSS Glossary 1.2
  - Central management of encryption keys, policy and reporting
  - Key Management - Industry Standards are missing (IEEE P1619.3, OASIS/KMIP …)

Data Protection – One-way vs. Two-way

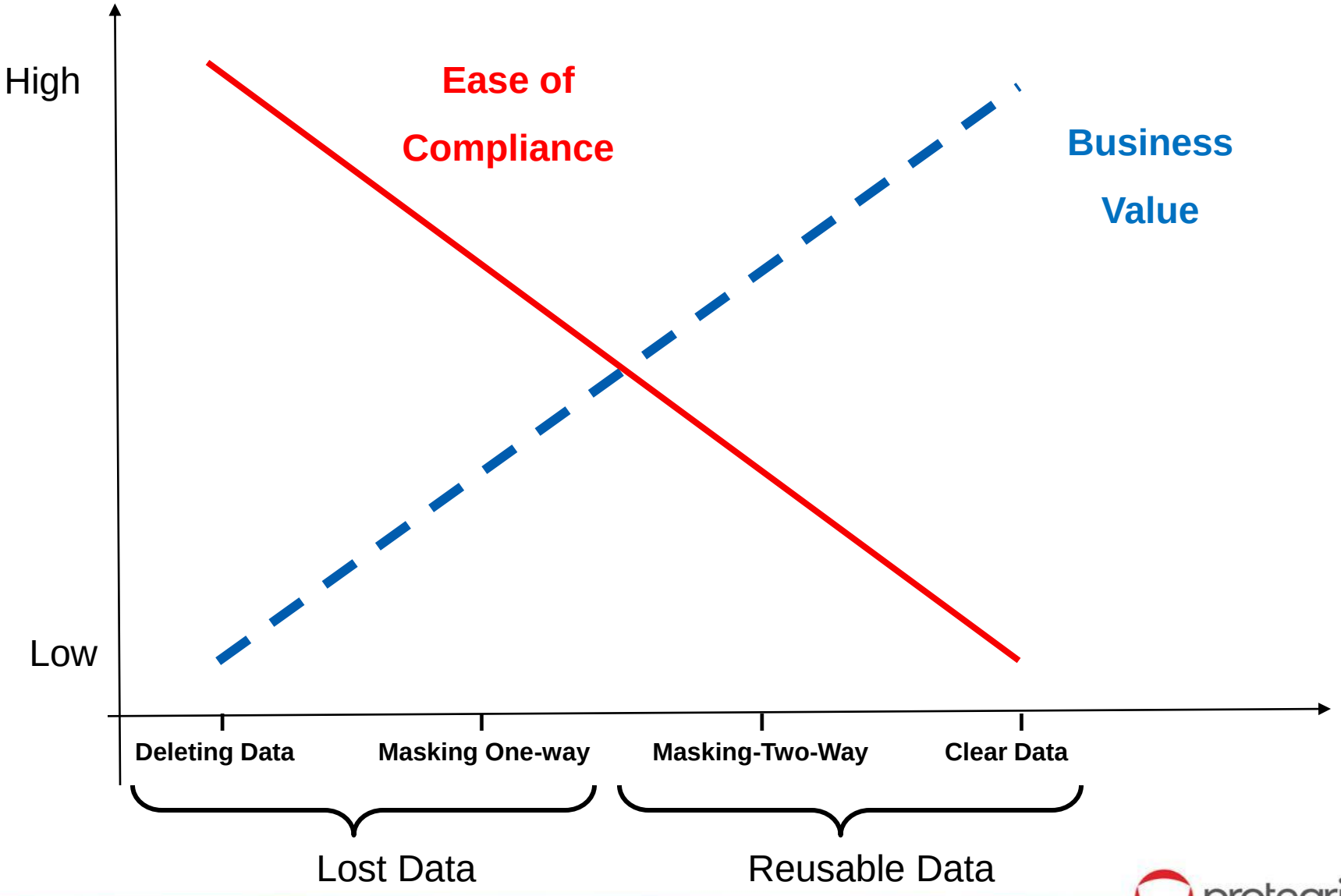


Unprotected sensitive information:

Protected sensitive information

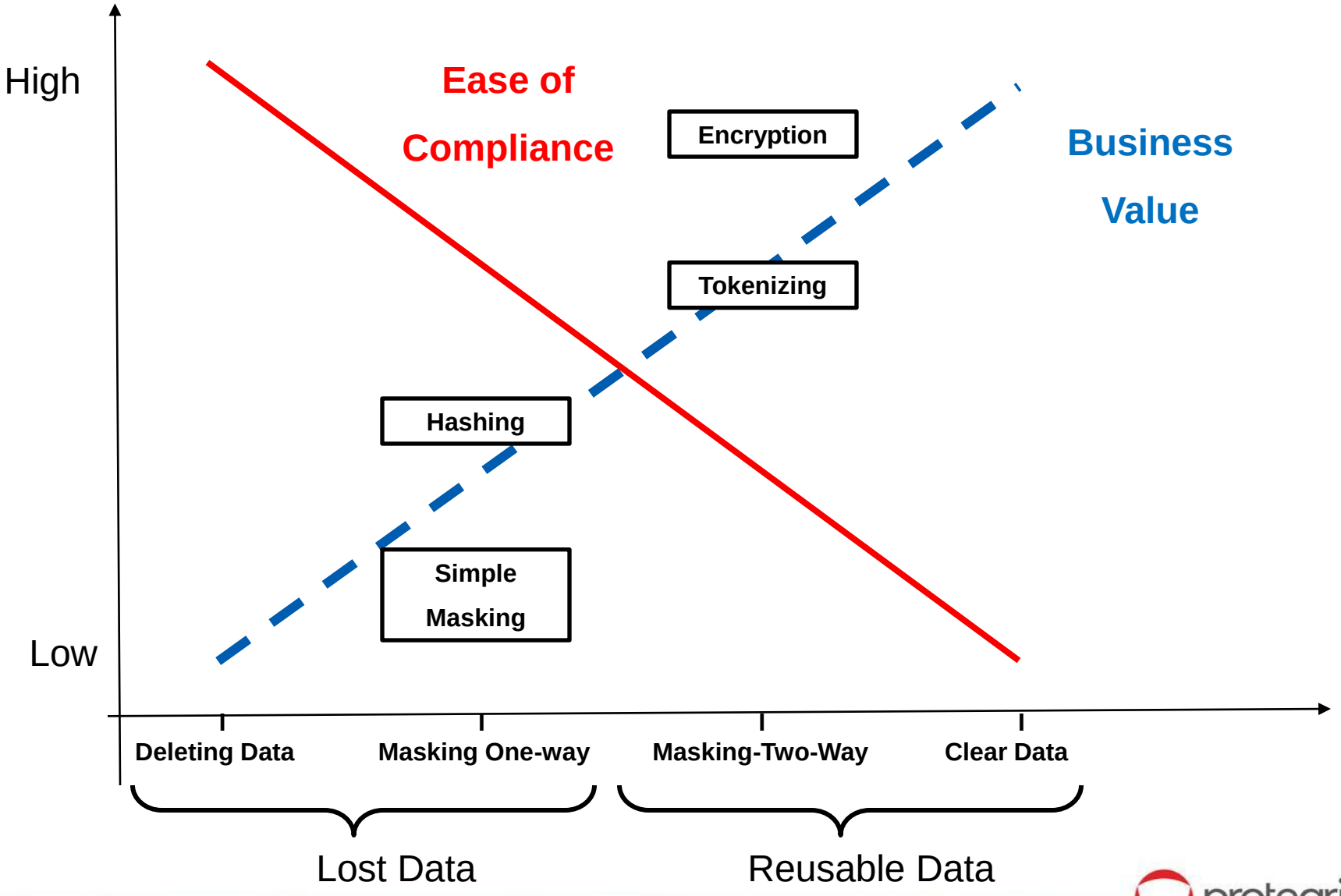
protegrity

Business Value vs. Ease of Compliance



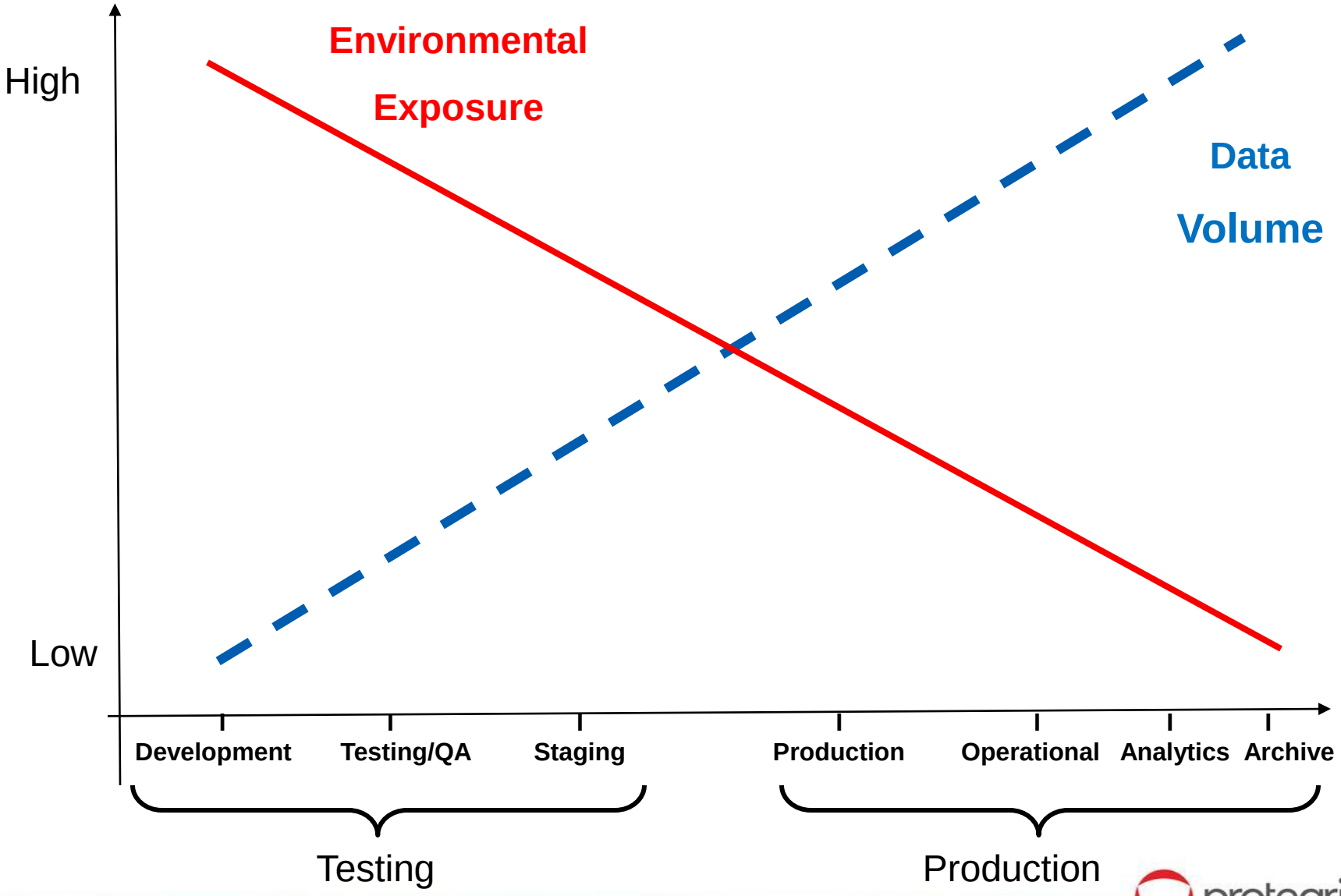
protegrity

Business Value vs. Ease of Compliance



protegrity

Data Volumes vs. Data Exposure



protegrity

Column Encryption Solutions – Some Considerations

Area of Evaluation	3rd Party	Oracle 9	Oracle 10 TDE	Oracle 11 TDE
Performance, manage UDT or views/triggers				
Support for both encryption and replication				
Support for Oracle Domain Index for fast search				
Keys are local; re-encryption if moving A -> B				
Separation of duties/key control vector				
Encryption format specified				
Data type support				
Index support beyond equality comparison				
HSM (hardware crypto) support				
HSM password not stored in file				
Automated and secure master key backup procedure				
Keys exportable				

Best Worst

protegrity