New York Oracle Users Group, Inc.

NYOUG

Get More for Less:

Enhance Data Security and Cut Costs

Ulf Mattsson, CTO, Protegrity Corporation

Dominic Dougherty, Protegrity Technical Support

Agenda

- PCI DSS and State Legislation
- Different data protection options for Oracle
- A risk adjusted methodology for determining appropriate solutions
- Case studies in protecting against internal and external threats
- Data encryption, enterprise key management, tokenization and database activity monitoring.
- Cost effective protection of data throughout the entire data flow

MasterCard
Academy of Risk Management

Payment System Integrity

MasterCard Worldwide

How to Evaluate Encryption Technologies

Ulf Mattsson, CTO

Protegrity

ISSA
Information Systems Security Association

New York Metro Chapter

Information | My Registration

Summary | Invitation | Agenda | Presenter Bios | Who should attend?

March 18, 2009: The Reality of PCI-DSS Compliance

ISSA New York Metro Chapter - Educational Program

Summary

The 2007 Computer Security Institute (CSI) Report indicates that more than one fifth of those surveyed have been victimized by a targeted attack. The study also concluded that financial fraud overtook virus attacks for the first time in seven years as the number one cause of financial losses from an IT security breach. Finally, customer and proprietary information was the second worst cause of financial loss. These trends show that the payment card industry faces more data security threats than ever before. The Payment Card Industry Data Security Standard (PCI-DSS) was created to mitigate these threats.

This session examines the challenges faced by organizations as they address their PCI DSS compliance requirements.

Presenter Bios

Ulf Mattsson, Protegrity Corporation

Ulf T. Mattsson, Chief Technology Officer, Protegrity Corporation, created the initial architecture of Protegrity's database security technology, for which the company owns several key patents. His extensive IT and security industry experience includes 20 years with IBM as a manager of software development and a consulting resource to IBM's Research and Development organization. He specializes in the areas of IT Architecture and IT Security. Ulf is the inventor of a number of European patents and US Patents in the areas of Encryption Key Management, Separation of Duties, Policy Driven Data Encryption, Internal Threat Protection, Data Usage Control, Dynamic Access Control, Intrusion Prevention and Cross System Layer Security. He holds a master's degree in physics, a degree in finance and a degree in electrical engineering.

protegrity

PCI Security Standards Council™

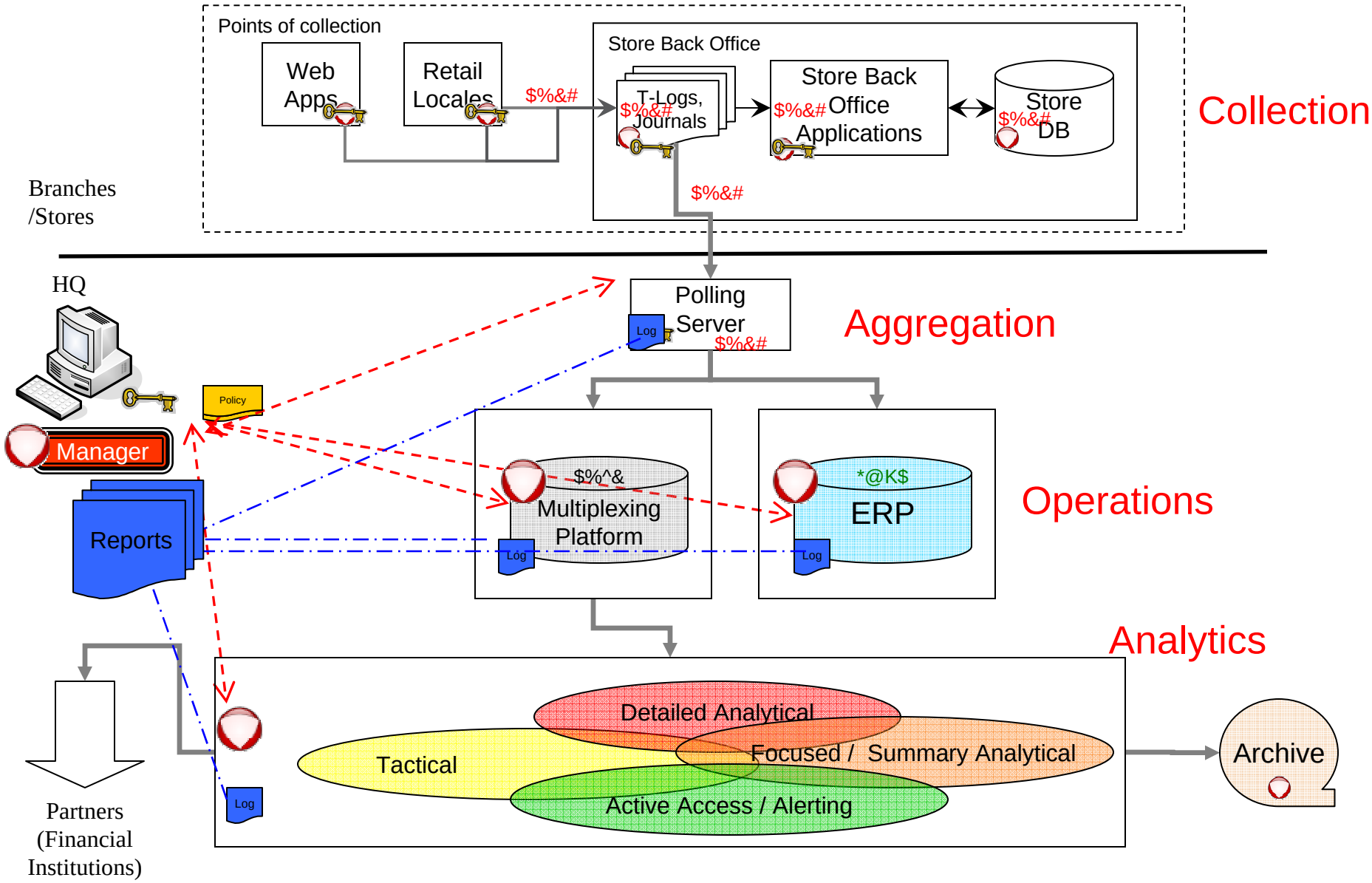
Participating Organization

http://www.knowpci.com

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.

Security for the Sensitive Data Flow



PCI DSS 3 - Protect Stored Cardholder Data

Section 3.4

- Render PAN, at minimum, unreadable anywhere it is stored (including on portable digital media, backup media, in logs) by using any of the following approaches:
  - One-way hashes based on strong cryptography
  - Truncation
  - Index tokens and pads (pads must be securely stored)
  - Strong cryptography with associated key-management processes and procedures
- The MINIMUM account information that must be rendered unreadable is the PAN.
- *Notes:*
  - *If for some reason, a company is unable render the PAN unreadable, refer to Appendix B:* Compensating Controls.
  - *"Strong cryptography" is defined in the PCI DSS Glossary of* Terms, Abbreviations, and Acronyms

Oracle and PCI DSS 3.4 - Protect Stored Cardholder Data

- Oracle Advanced Security Transparent Data Encryption (TDE) can be used to encrypt the number on media and backup.
- Optionally TDE can be used with Oracle RMAN to encrypt the entire backup when backed up to disk.
- Oracle Secure Backup provides a solution for backing up and encrypting directly to tape storage.
- Encryption algorithms supported include AES and 3DES with 128, 192 (default), or 256 bit key length.
- Oracle Advanced Security Transparent Data Encryption (TDE) has key management built-in.
- Encrypted column data stays encrypted in the data files, undo logs, and redo logs, as well as in the buffer cache of the system global area (SGA). SHA-1 and MD5 are used for integrity.

PCI DSS 3.5 & 3.6 - Protect Encryption Keys

- *"Protect encryption keys used for encryption of cardholder data against both disclosure and misuse.*
  - *3.5.1 Restrict access to keys to the fewest number of custodians necessary*
  - *3.5.2 Store keys securely in the fewest possible locations and forms."*
- *"Fully document and implement all key management processes and procedures for keys used for encryption of cardholder data, including the following:*
  - *3.6.1 Generation of strong keys, secure key distribution, secure key storage*
  - *3.6.4 Periodic changing of keys*
  - *• As deemed necessary and recommended by the associated application (for example, re-keying); preferably automatically. At least annually.*
  - *3.6.5 Destruction of old keys*
  - *3.6.6 Split knowledge and establishment of dual control of keys (so that it requires two or three people, each knowing only their part of the key, to reconstruct the whole key)*
  - *3.6.7 Prevention of unauthorized substitution of keys*
  - *3.6.8 Replacement of known or suspected compromised keys*
  - *3.6.9 Revocation of old or invalid keys*

Oracle and PCI DSS 3.5 - Protect Encryption Keys

- Oracle Advanced Security Transparent Data Encryption (TDE) keys are stored in the database and encrypted using a separate master key that is stored in the Oracle Wallet, a PKCS#12 file on the operating system.

- The Oracle Wallet is encrypted using the wallet password; in order to open the wallet from within the database requires the 'alter system' privilege.

- Oracle Database Vault command rules can be implemented to further restrict who, when, and where the 'alter system' privilege can be executed.

Data Protection Options

- Data Stored As
  - Clear – actual value is readable
  - Hash – unreadable, not reversible
  - Encrypted – unreadable, reversible
  - Replacement value (tokens) – unreadable, reversible
  - Partial encryption/replacement – unreadable, reversible

Data Protection Options

- Hash
  - Non – reversible
  - Strong protection
    - Keyed hash (HMAC)
    - Unique value if salt is used
- Advantages
  - None really
- Considerations
  - Key rotation for keyed hash
  - Size and type
  - Transparency

Data Protection Options

- Strong Encryption
  - Industry standard (AES CBC …)
  - Highest security level
- Advantages
  - Widely deployed
  - Compatibility
  - Performance
- Considerations
  - Storage and type
  - Transparency to applications
  - Key rotation

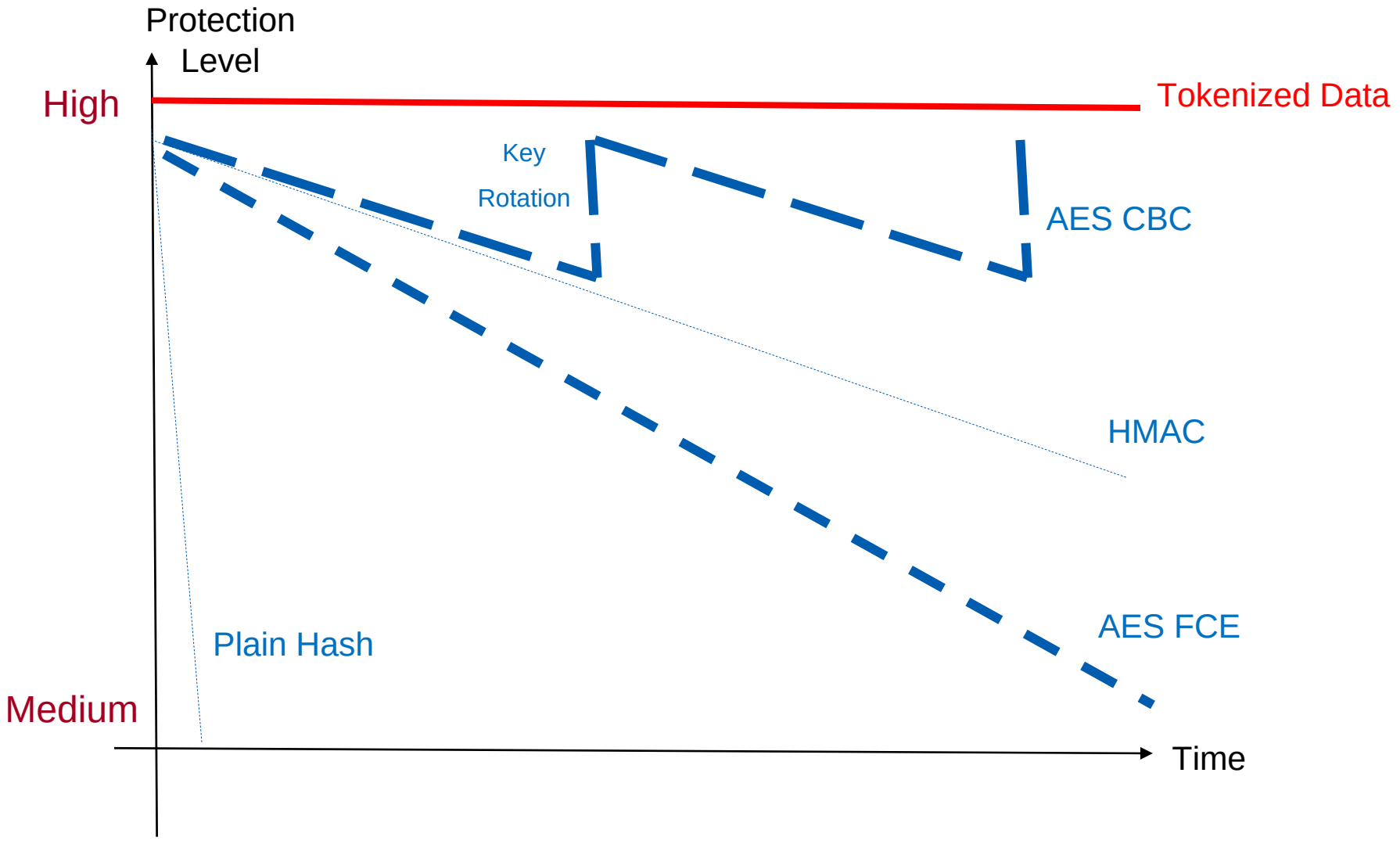
Data Protection Options

- Format Controlling Encryption
  - Maintains data type, length
- Advantages
  - Reduces changes to downstream systems
  - Storage
  - Partial encryption
- Considerations
  - Performance
  - Security
  - Key rotation
  - Transparency to applications

Data Protection Options

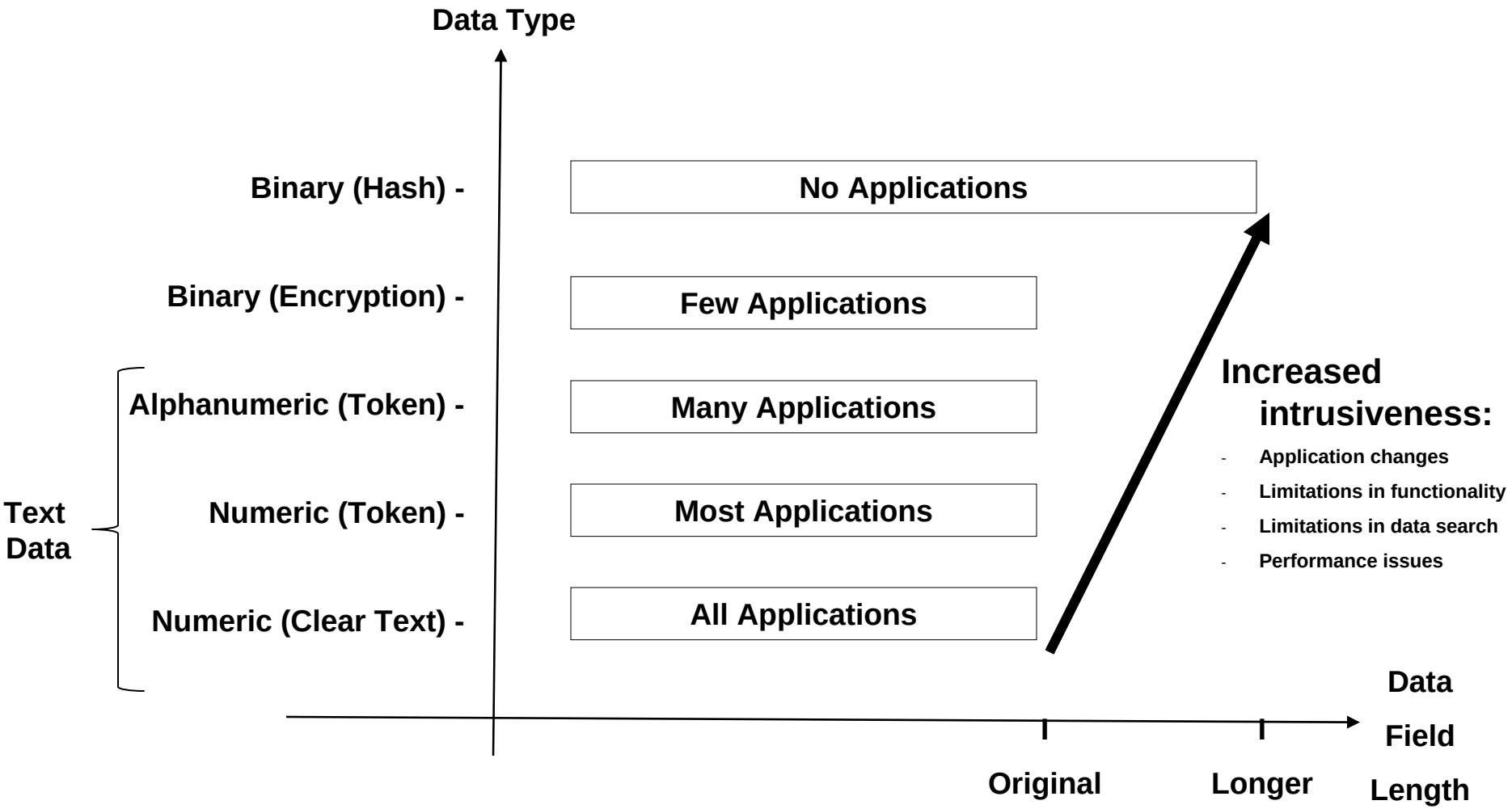
- Replacement Value (i.e. tokens, alias)
  - Proxy value created to replace original data
  - Centrally managed, protected
- Advantages
  - No changes to most downstream systems
  - Out of scope for compliance
  - No local key rotation
  - Partial replacement
- Considerations
  - Transparency for applications needing original data
  - Availability and performance for applications needing original data

Field Level Data Protection Methods vs. Time



protegrity

Applications typically react to a Break in the Data Field Format



This is a generalized example

Data Protection Capabilities

Storage	Performance	Storage	Security	Transparency
Clear	●	●	○	●
Strong Encryption	◔	◔	●	◓
Format Controlling Encryption	◓	●	◓	◔
Token	◔	●	●	◔
Hash	●	◔	●	○

Best ● ◔ ◓ ◔ ○ Worst

protegrity

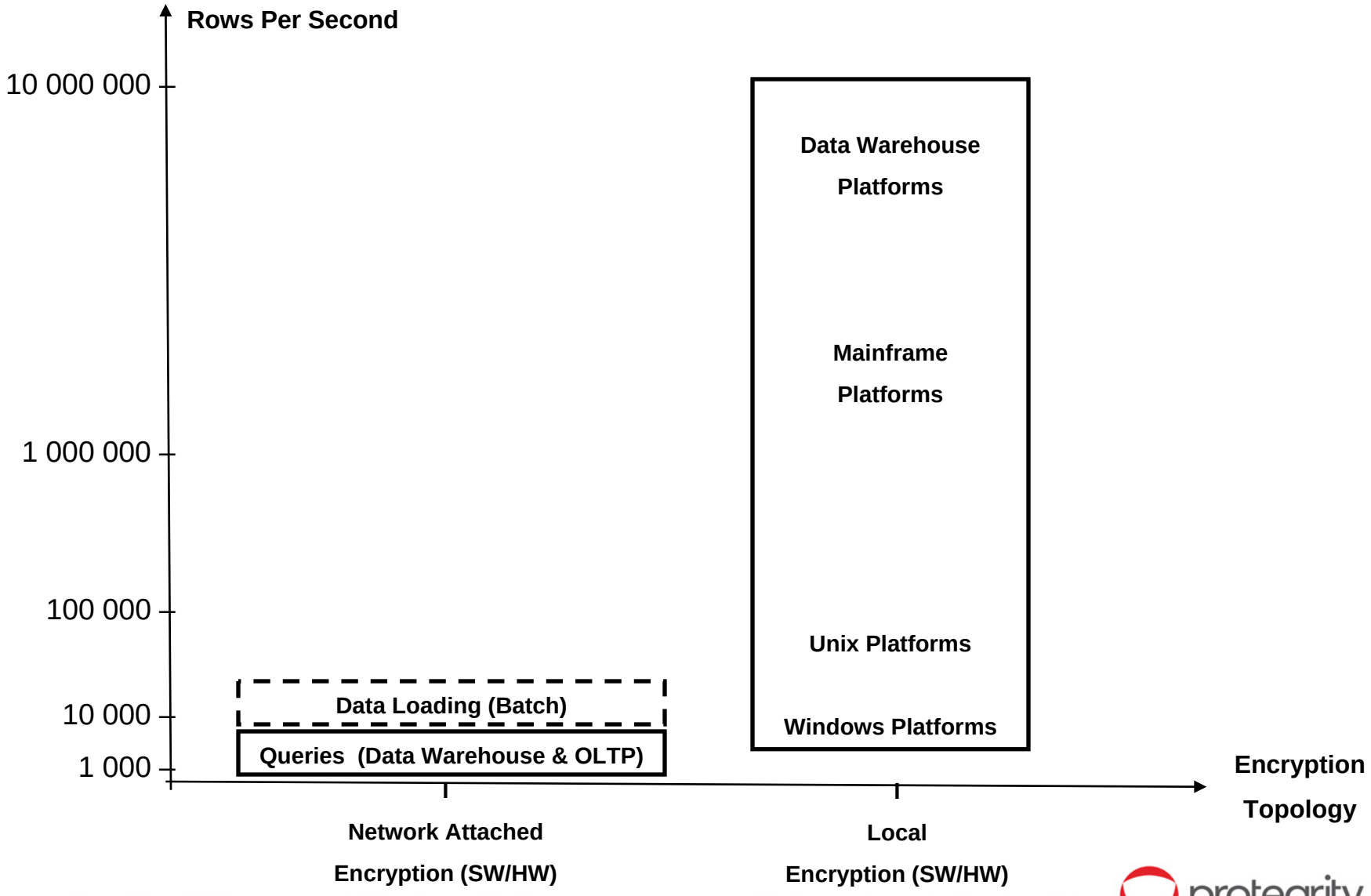
Data Protection Implementation Choices

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

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

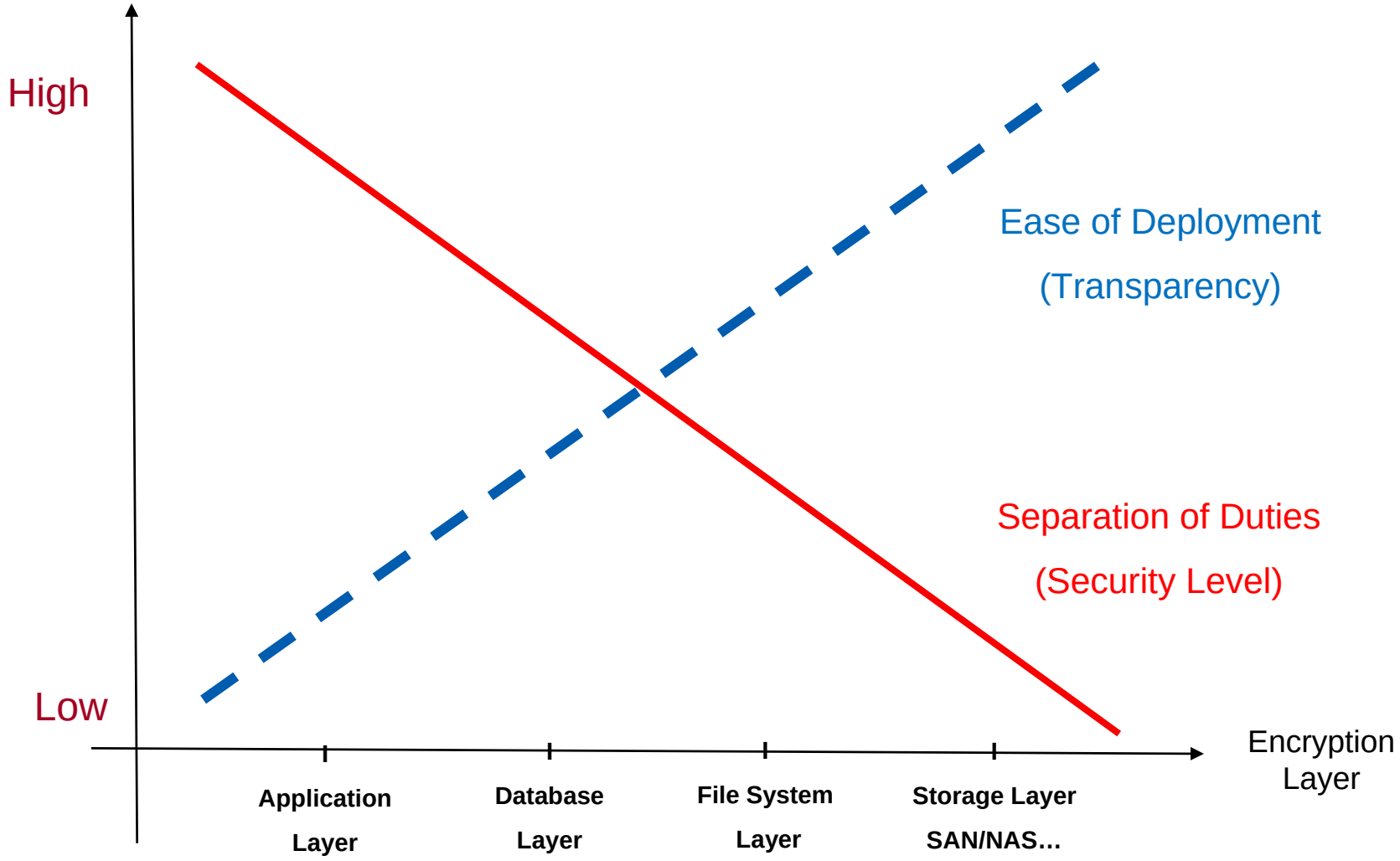
Best ● ◔ ◒ ◔ ○ Worst

Column Encryption Performance - Different Topologies



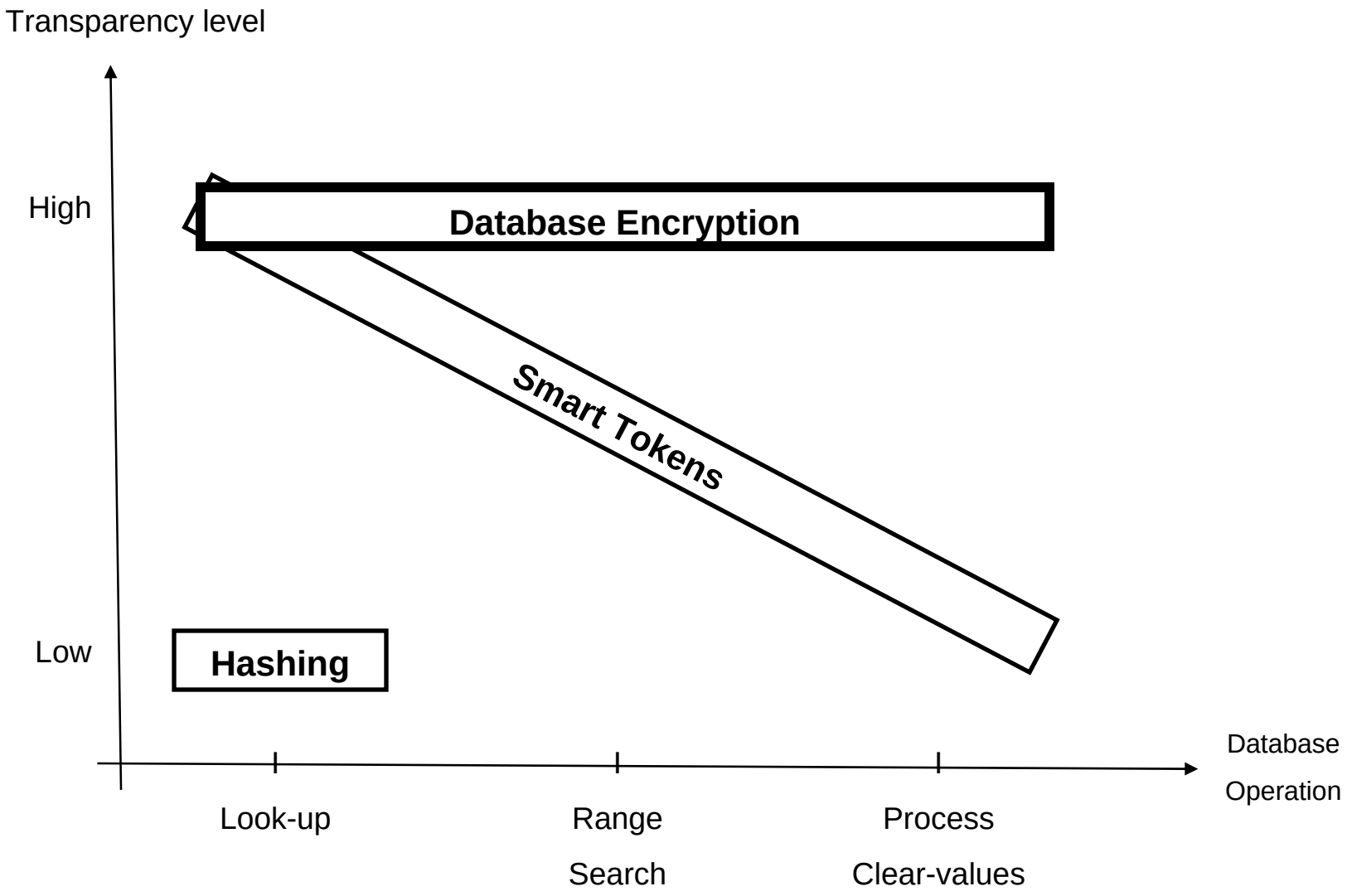
protegrity

Generalization: Encryption at Different System Layers

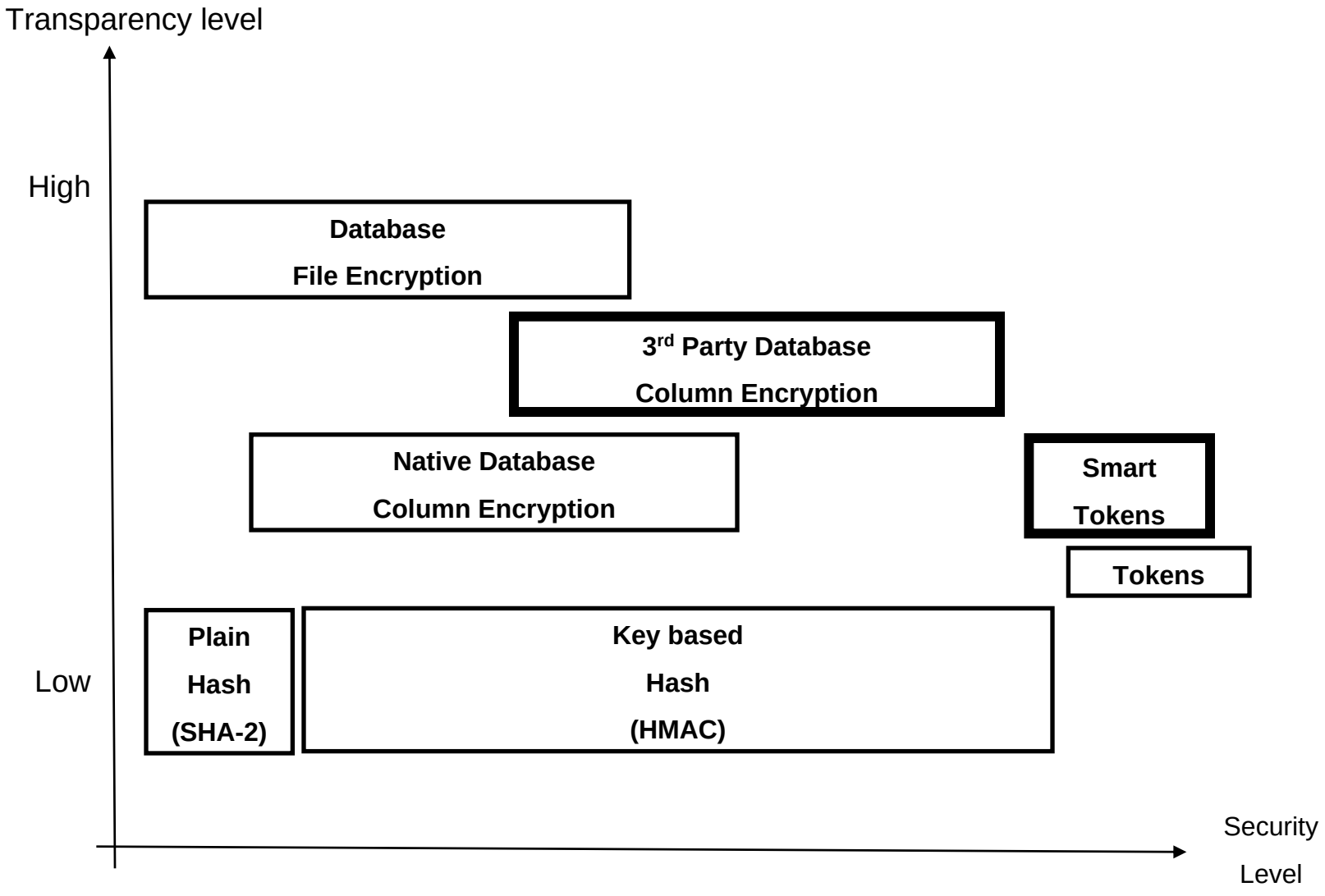


protegrity

Application Transparency – Encryption, Tokens & Hashing



Application Transparency



protegrity

Oracle Implementations – Customer Requirements

- Protect data
- Protect data and audit activity
- Protect data, audit activity and grant/revoke User access
- Protect data, audit activity, grant/revoke user access and restrict access to specific IP's
- Protect data, audit activity, grant/revoke user access ,restrict access to specific IP's and transfer secure data to Development and Test systems
- All of the above and Secure Backups

Oracle Implementations – Tools

- Oracle Data Vault
- Oracle Audit Vault
- Oracle Data Masking
- Oracle Secure Backup
- Oracle Transparent Data Encryption
- Oracle Wallet
- Oracle VPD
  - Oracle Fine Grained Access Control
  - Oracle Row Level Security (DBMS_RLS)
- Oracle Proxy Authentication
- Oracle Advanced Security
  - Oracle Data Encryption API (dbms_crypto)
  - Network Encryption
  - Strong Authentication
- Oracle Application Context
- Oracle Label Security
- Oracle Fine Grained Auditing (dbms_fga)

Oracle Implementations – pre requisites

- Implementations of any solution cannot be 100% successful without consulting with Business Owners, Auditors, System Administrator , DBA and End users.
- Encryption Vendors need to understand data flows before recommending solution as one solution will not fit all requirements.

Oracle Implementation - Sample code

Using DBMS_CRYPTO (Oracle 10G+)

```
declare
        input_string VARCHAR2 (200) := 'Secret Message';
        output_string VARCHAR2 (200);
        encrypted_raw RAW (2000); -- stores encrypted binary text
        decrypted_raw RAW (2000); -- stores decrypted binary text
        num_key_bytes NUMBER := 256/8; -- key length 256 bits (32 bytes)
        key_bytes_raw RAW (32); -- stores 256-bit encryption key encryption_type
        PLS_INTEGER := -- total encryption type
        DBMS_CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC + DBMS_CRYPTO.PAD_PKCS5;
begin
        DBMS_OUTPUT.PUT_LINE ('Original string: ' || input_string);
        key_bytes_raw := DBMS_CRYPTO.RANDOMBYTES (num_key_bytes);
        encrypted_raw := DBMS_CRYPTO.ENCRYPT ( src => UTL_I18N.STRING_TO_RAW (input_string, 'AL32UTF8'), typ =>
        encryption_type, key => key_bytes_raw ); -- The encrypted value in the encrypted_raw variable can be used here
        decrypted_raw := DBMS_CRYPTO.DECRYPT ( src => encrypted_raw, typ => encryption_type, key => key_bytes_raw );
        output_string := UTL_I18N.RAW_TO_CHAR (decrypted_raw, 'AL32UTF8');
        DBMS_OUTPUT.PUT_LINE ('Decrypted string: ' || output_string);
end;
```

Oracle Implementation - Sample code

Oracle Advanced Security : TDE+Oracle Wallet

Tablespace Encryption

- ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "myPassword";
- CREATE TABLESPACE encrypted_ts DATAFILE '/u01/app/oracle/oradata/DB11G/encrypted_ts01.dbf' SIZE 128K AUTOEXTEND ON NEXT 64K ENCRYPTION USING 'AES256' DEFAULT STORAGE(ENCRYPT);
- CREATE TABLE ets_test (id NUMBER(10), data VARCHAR2(50)) TABLESPACE encrypted_ts;
- CREATE INDEX ets_test_idx ON ets_test(data) TABLESPACE encrypted_ts;

Column Level Encryption

- CREATE TABLE CUST_PAYMENT_INFO (FNAME VARCHAR2(11), LNAME VARCHAR2(10), ORDERNO NUMBER(5),
- CC_NUMBER VARCHAR2(16) ENCRYPT NO SALT, EXP_DT CHAR(4));
- ALTER TABLE CUST_PAYMENT_INFO *REKEY* USING 'AES192'

Oracle Implementation - Sample code

Application Contexts

- CREATE CONTEXT Order_entry USING Apps.Oe_ctx;
- CREATE OR REPLACE PACKAGE apps.oe_ctx AS

```
        PROCEDURE set_cust_num ;
    END;
```

- CREATE OR REPLACE PACKAGE BODY apps.oe_ctx

```
        AS PROCEDURE set_cust_num IS custnum NUMBER;
    BEGIN
        SELECT cust_no INTO custnum FROM custs
        WHERE uname = SYS_CONTEXT('USERENV', 'session_user');
        /* SET cust_num attribute in 'order_entry' context */
        DBMS_SESSION.SET_CONTEXT('order_entry', 'cust_num', custnum);
        DBMS_SESSION.SET_CONTEXT('order_entry', 'cust_num', custnum);
    END set_cust_num;
    END;
```

- CREATE VIEW ORDERS AS

```
    SELECT * FROM Orders_tab WHERE Custno = SYS_CONTEXT('order_entry', 'cust_num') ;
```

Oracle Implementation - Sample code

Using dbms_FGA

- BEGIN
  dbms_fga.add_policy
  (object_schema=>'SALES',
  object_name=>'PAYMENT',
  policy_name=>'PAYMENT_ACCESS',
  audit_column => 'CARD_NUM',
  statement_types => 'UPDATE, DELETE, SELECT',
  audit_condition => 'CARD_NUM IS NOT NULL');
  END;

- select policy_name, object_name, object_schema, policy_text, policy_column from dba_audit_policies;
- select * from payment;
- select card_num from payment;
- select timestamp, db_user, os_user,object_schema, object_name,sql_text from dba_fga_audit_trail;

Oracle Implementation - Sample code

Securing Oracle Listener

- **Protocol.ora**

  tcp.validnode_checking = YES

  tcp.excluded_nodes = (finance.myco.uk, mktg.myco.us, 123.12.1.0)

  tcp.invited_nodes = (services.myco.fr, csr.myco.es)

SQLNET.ORA

- #ASO Encryption

  sqlnet.encryption_server=accepted

  sqlnet.encryption_client=requested

  sqlnet.encryption_types_server=(RC4_40)

  sqlnet.encryption_types_client=(RC4_40)
- Oracle Advanced Security Network Data Integrity
- #ASO Checksum

  sqlnet.crypto_checksum_server=requested

  sqlnet.crypto_checksum_client=requested

  sqlnet.crypto_checksum_types_server = (MD5)

  sqlnet.crypto_checksum_types_client = (MD5)

Oracle Implementation – Security

- Security can be as granular as needed, However, there is cost involved. (time, effort)
- Too much of Security can hamper performance
- Security can also hamper day to day administration of System
- Security mostly likely is introduced when the SDLC is completed or when the system is already in production. Customers should expect changes in the way the use the system if the security layer is added later on. (application, DB, Queries)
- Understand Data Flow helps to implement a better solution
- Solution when restored should have some kind of addition authorization requirement to ensure that backup cannot be recovered without intervention.

Oracle Implementation – Performance

- Stay away from encrypting any searchable columns
- Stay away from protecting column who have referenced integrity
- Implement Oracle technologies like Function Based Index and Domain Based Index.
- Revisit Data, change Data model to remove SSN, CCN, Account number from Primary keys, foreign keys.
- Have a look at the application for the possibility to change the way users Query data.
- Revisit data, Ensure that sensitive data resided only when it is necessary and not distributed on all systems. (data Masking)
- Classify data category to ensure data can be access only by users who need to see it (Label Security)
- Data Protection can be done at the Application level, Database level and OS level (file/disk) level. To provide a solution you would need to have a blend of all the above as well as network protection.

Oracle Implementation – Performance

- **Function Based Index**

  CREATE INDEX income_ix ON employees(salary + (salary*commission_pct));

  SELECT first_name||' '||last_name "Name" FROM employees WHERE (salary*commission_pct) + salary > 15000;

- **Domain Based Index**

  CREATE INDEX myidx ON mytable(docs) indextype is ctxsys.context;

  SELECT * from mytable where contains(docs,'some words') > 0;

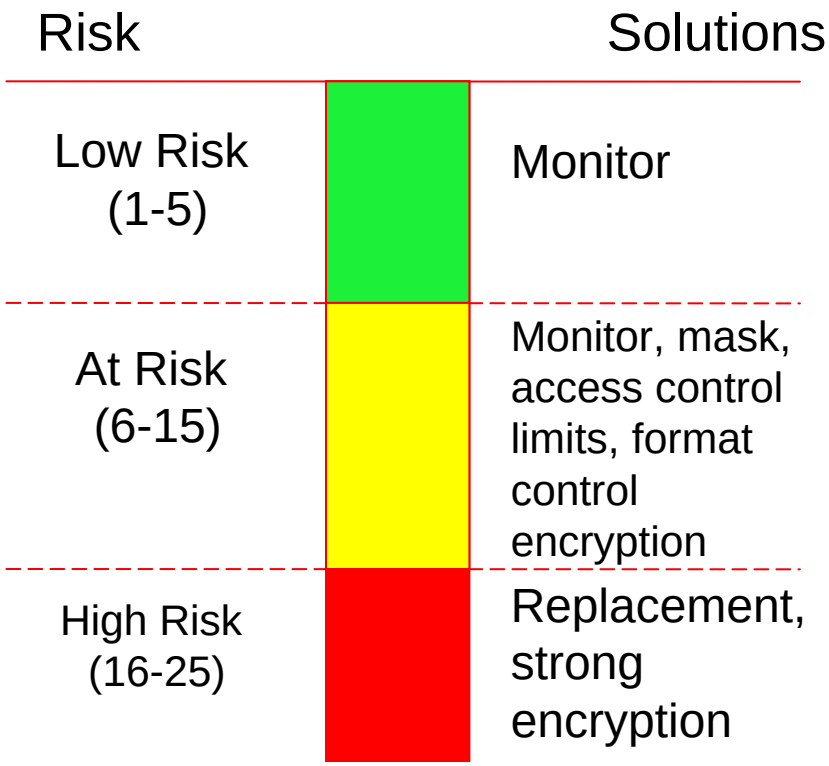
Oracle Implementation – Administration

- Solution should have Seperation of Duties
- Solution should be easy deployable, no addition special requirements should be needed (additional hardware, network configuration)
- Solution should have a mechanism of backup and recovery.

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



Operation Cost Factors

- Performance
  - Impact on operations - end users, data processing windows
- Storage
  - Impact on data storage requirements
- Security
  - How secure Is the data at rest
  - Impact on data access – separation of duties
- Transparency
  - Changes to application(s)
  - Impact on supporting utilities and processes

Data Security Management

- An integral part of technical and business process
- Security Policy
  - Centralized control of security policy
  - Consistent enforcement of protection
  - Separation of duties
- Reporting and Auditing
  - Compliance reports
  - Organization wide security event reporting
  - Alerting
  - Integration with SIM/SEM
- Key Management

Protecting Data in the Enterprise Data Flow

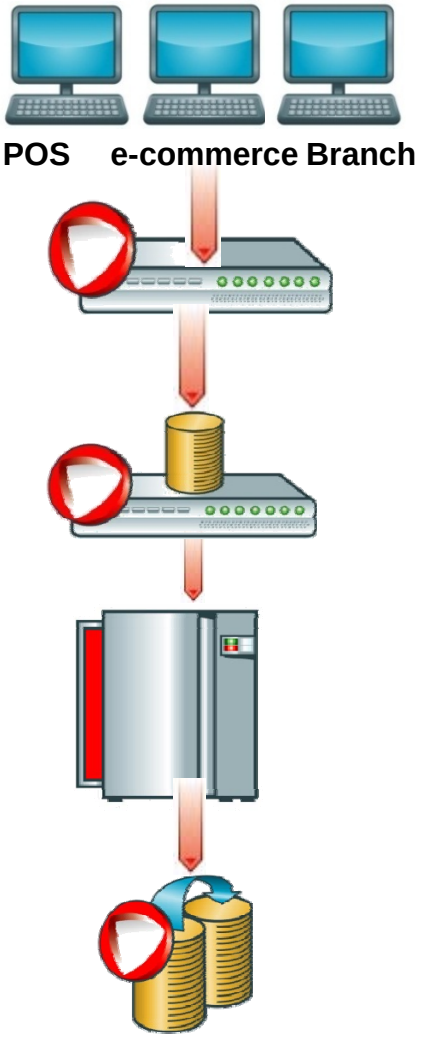
Collection

Aggregation

Operations

Analysis

Archive



- **'Information in the wild'**
  - **Short lifecycle / High risk**
- **Temporary information**
  - **Short lifecycle / High risk**
- **Operating information**
  - **Typically 1 or more year lifecycle**
  - **Broad and diverse computing and database environment**
- **Decision making information**
  - **Typically multi-year lifecycle**
  - **Homogeneous computing environment**
  - **High volume database analysis**
- **Archive**
  - **Typically multi-year lifecycle**
  - **Preserving the ability to retrieve the data in the future is important**

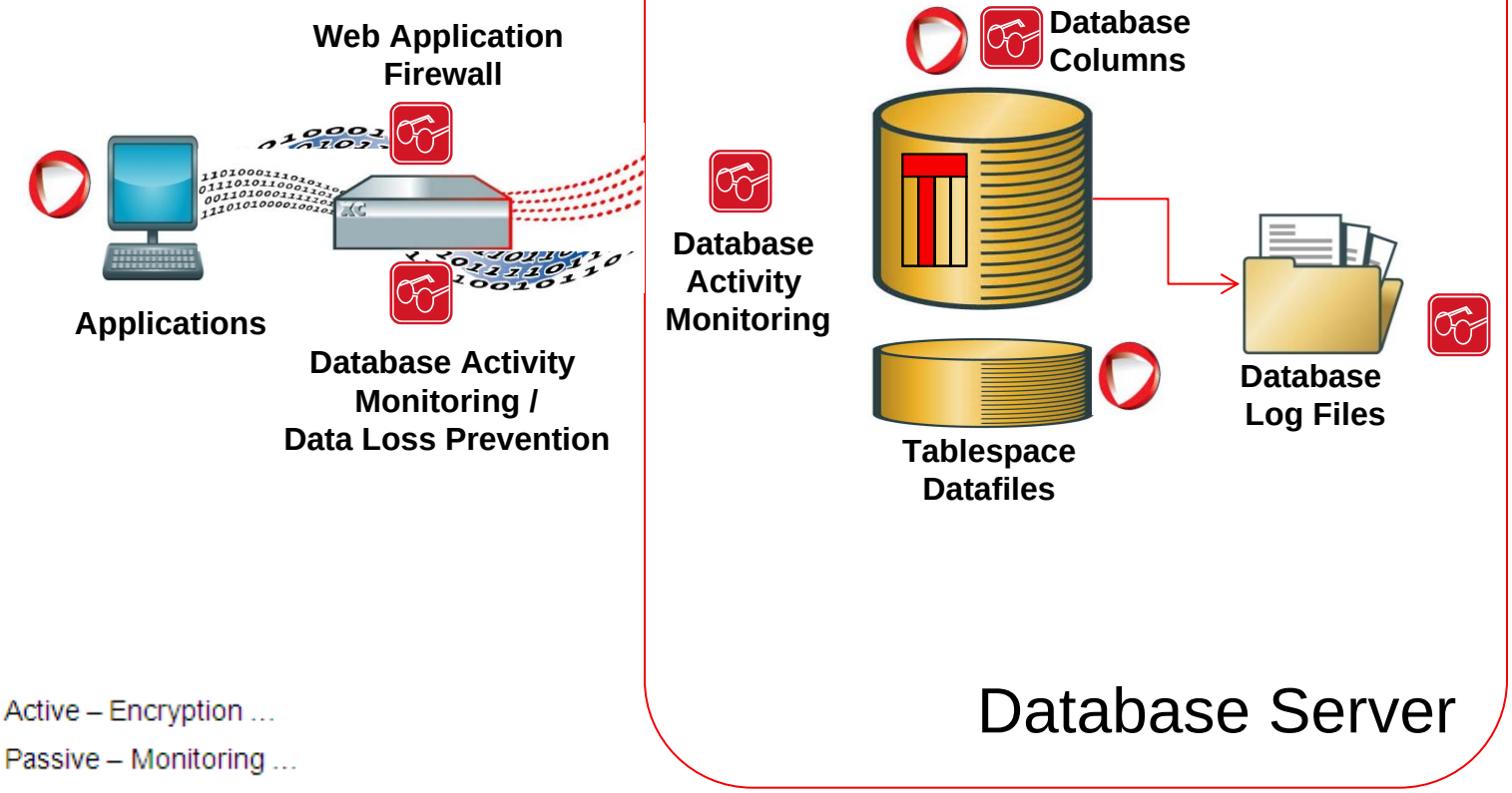
protegrity

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 …)

Protecting Data in the Enterprise Data Flow

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



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

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

Data Protection Options

Storage	Performance	Storage	Security	Transparency
Clear	●	●	○	●
Strong Encryption	◔	◔	●	◒
Format Controlling Encryption	◒	●	◒	◔
Token	◔	●	●	◔
Hash	●	◔	●	○

Best ● ◔ ◒ ◔ ○ Worst

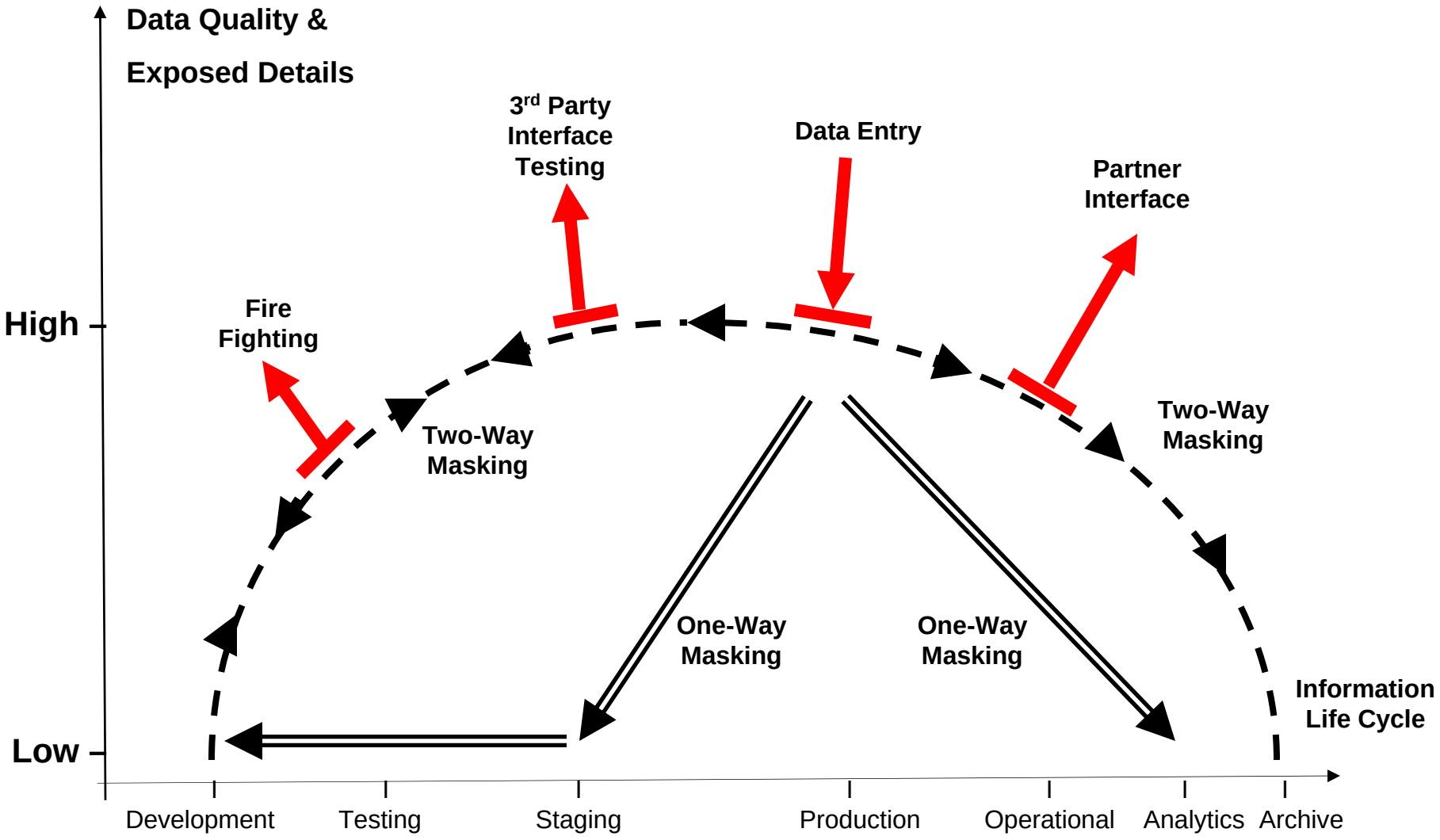
Data Protection Implementation Choices

System Layer	Performance	Transparency	Security
Application			
Database			
File System			

Topology	Performance	Scalability	Security
Local Service			
Remote Service			

Best  Worst

Data Masking – One-way vs. Two-way



Unprotected sensitive information:

Protected sensitive information

protegrity

Questions?

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

protegrity

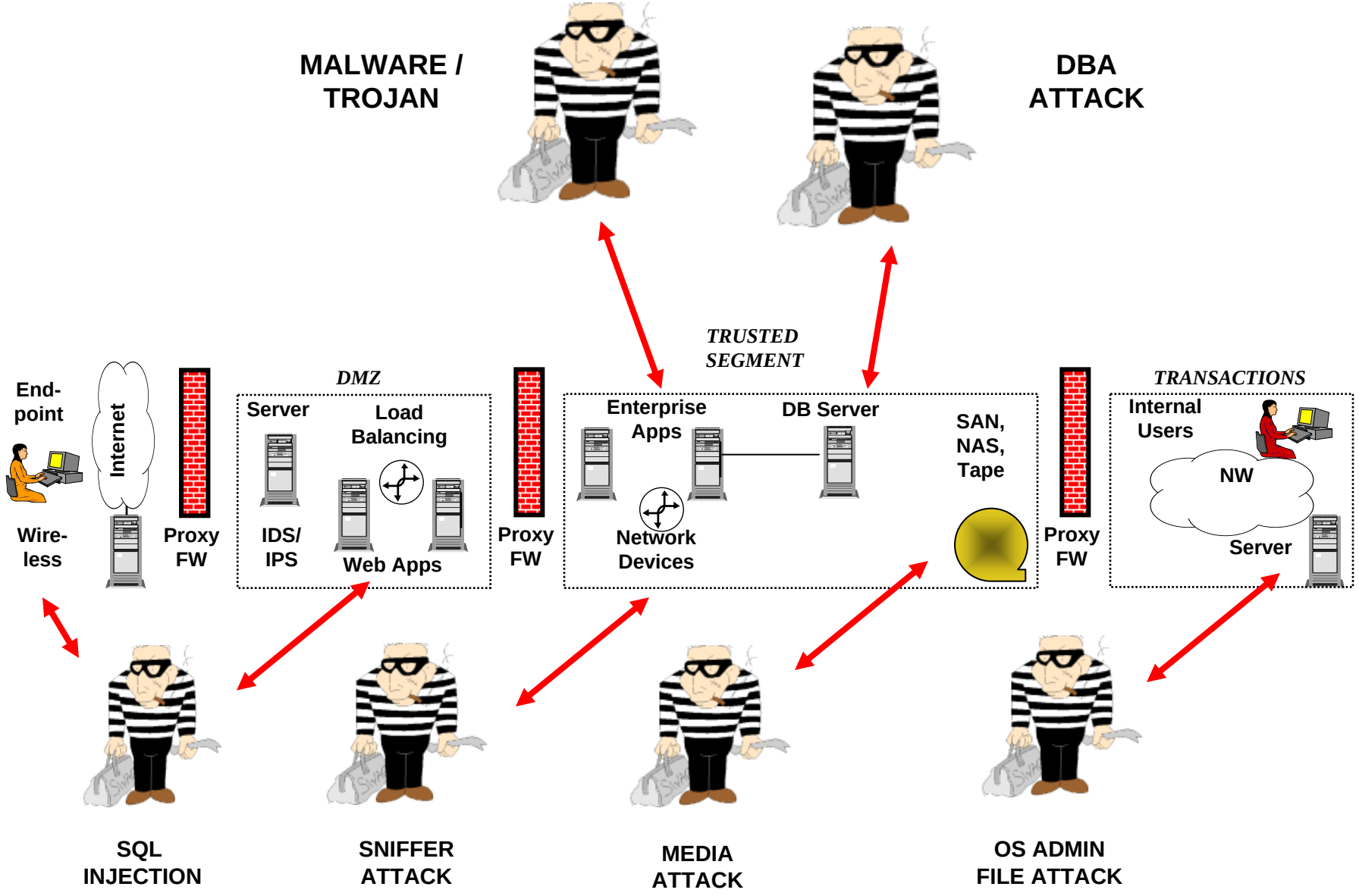
protecting your data.
protecting your business.

Appendix

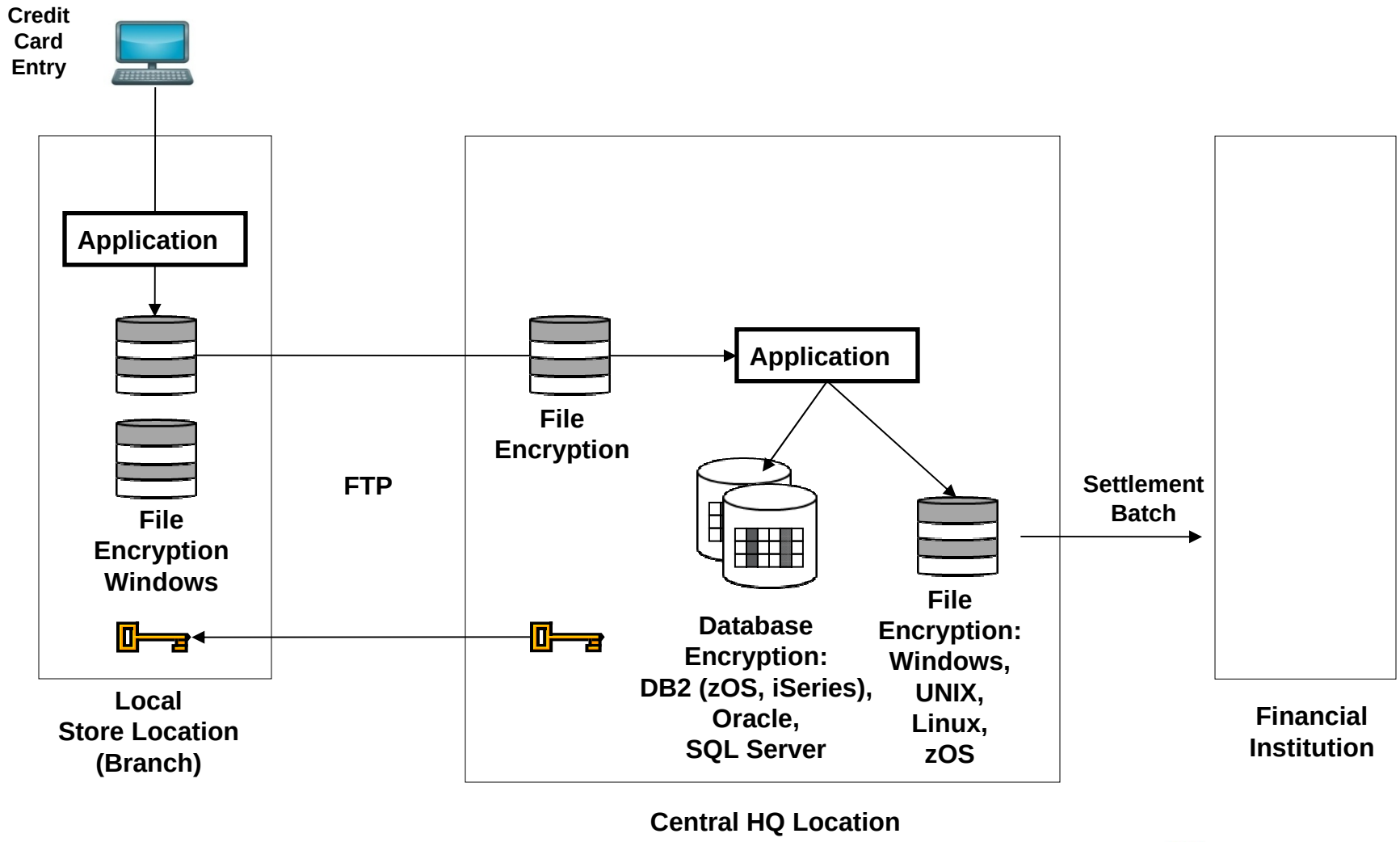
protegrity

protecting your data.
protecting your business.

Data Level Attacks

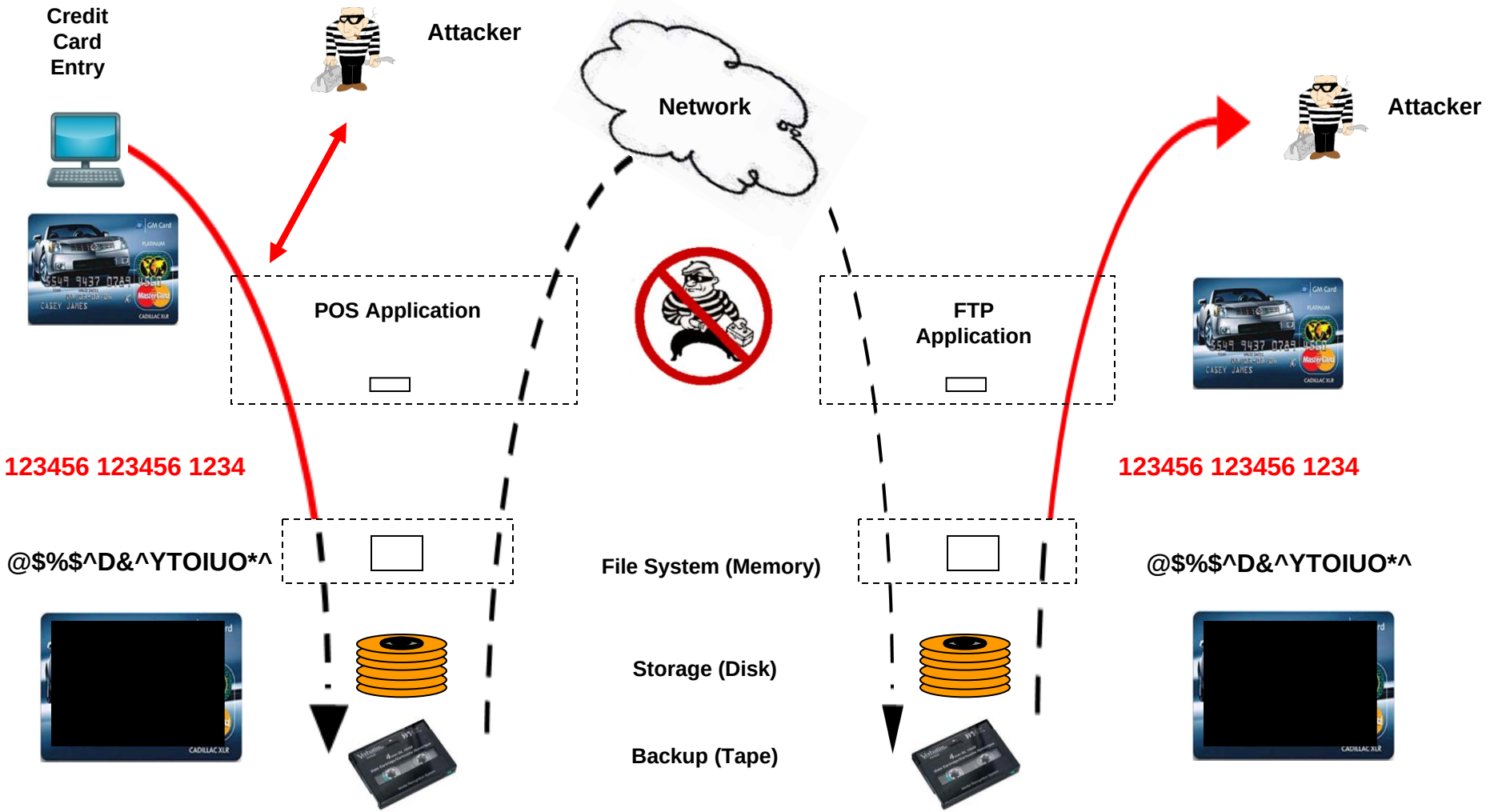


Case 1: Goal – PCI Compliance & Application Transparency

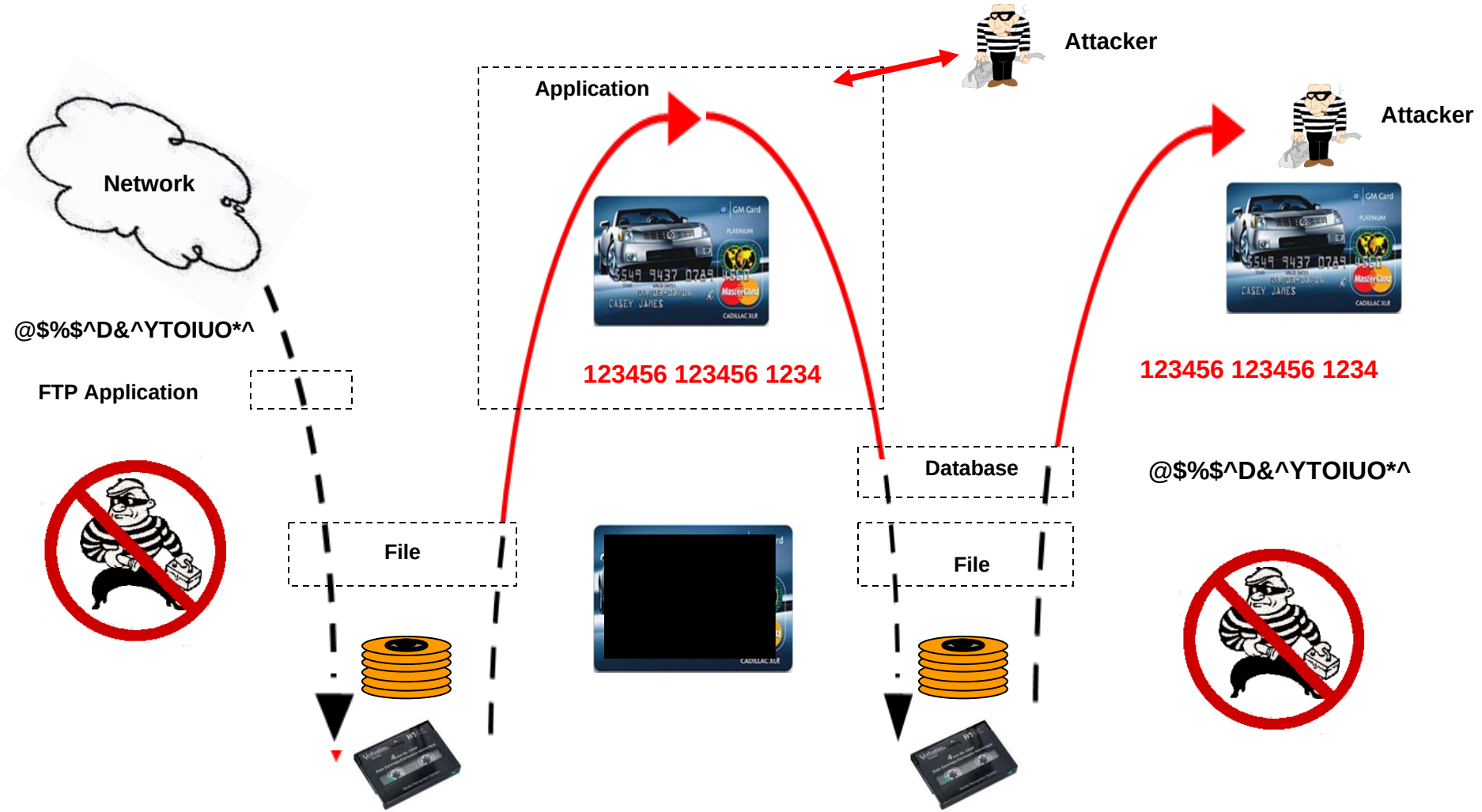


protegrity

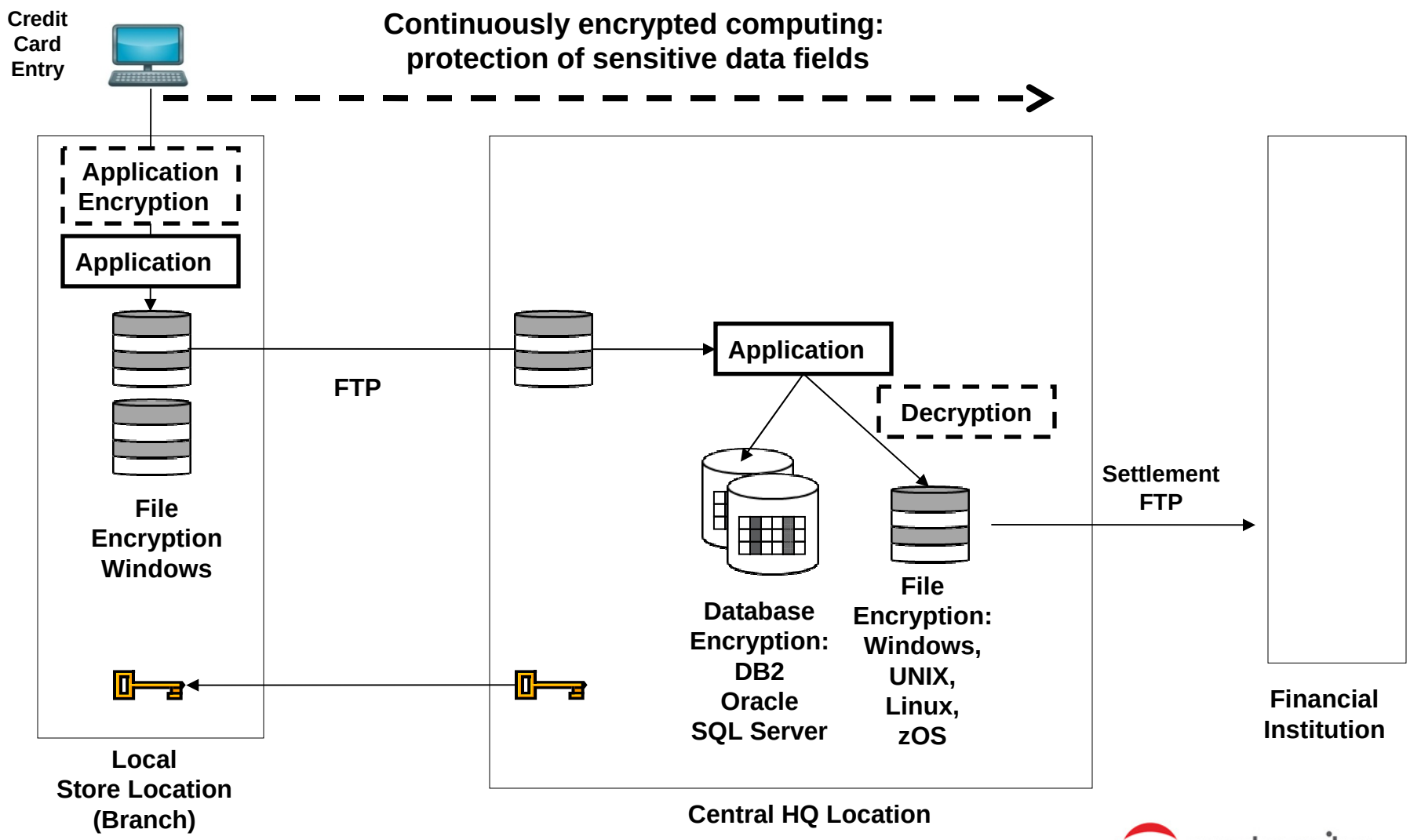
Case 1: File Encryption & FTP



Case 1: From Encrypted File to Encrypted Database

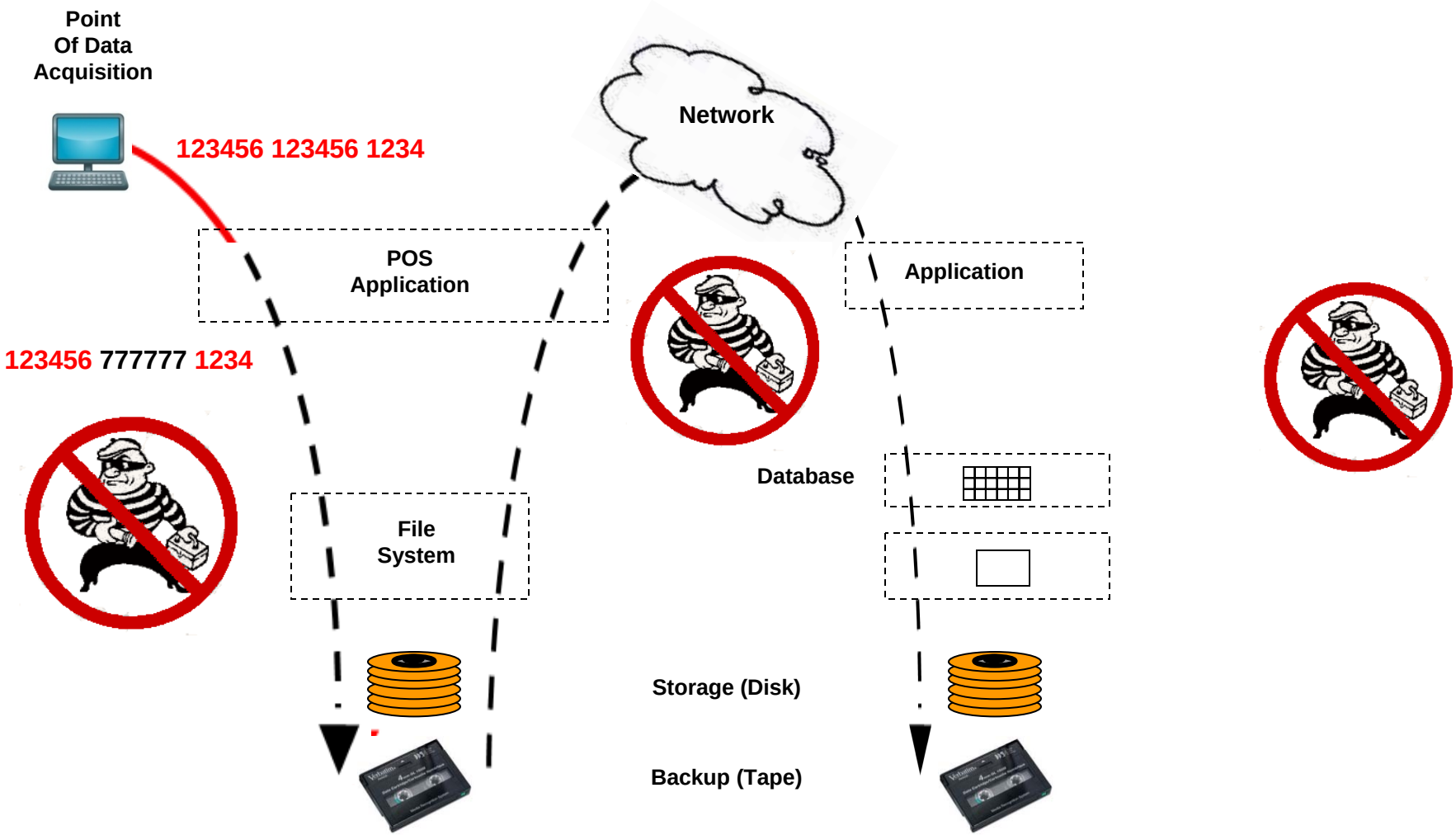


Case 2a: Goal – Addressing Advanced Attacks & PCI

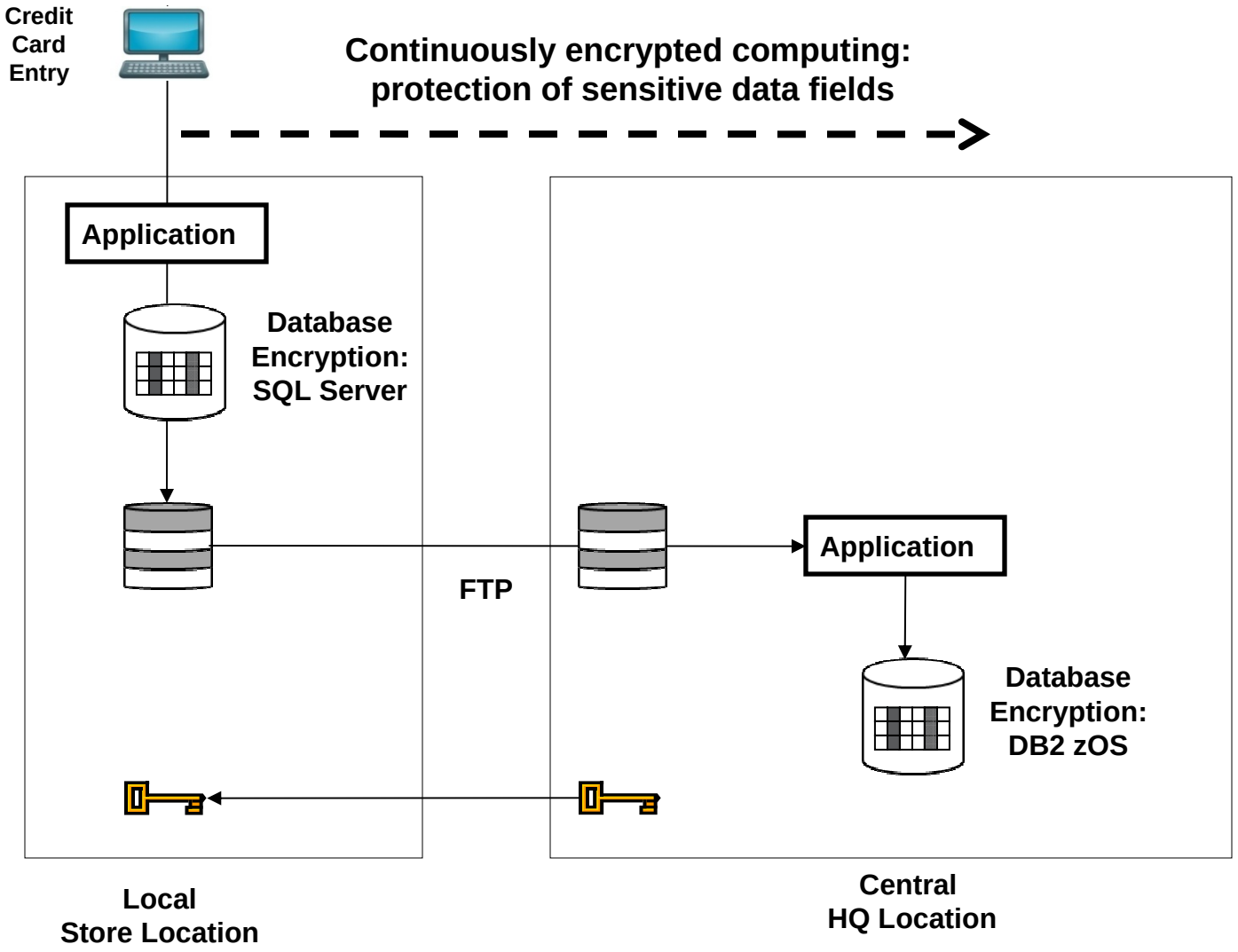


protegrity

Case 2a: Application Encryption to Encrypted Database

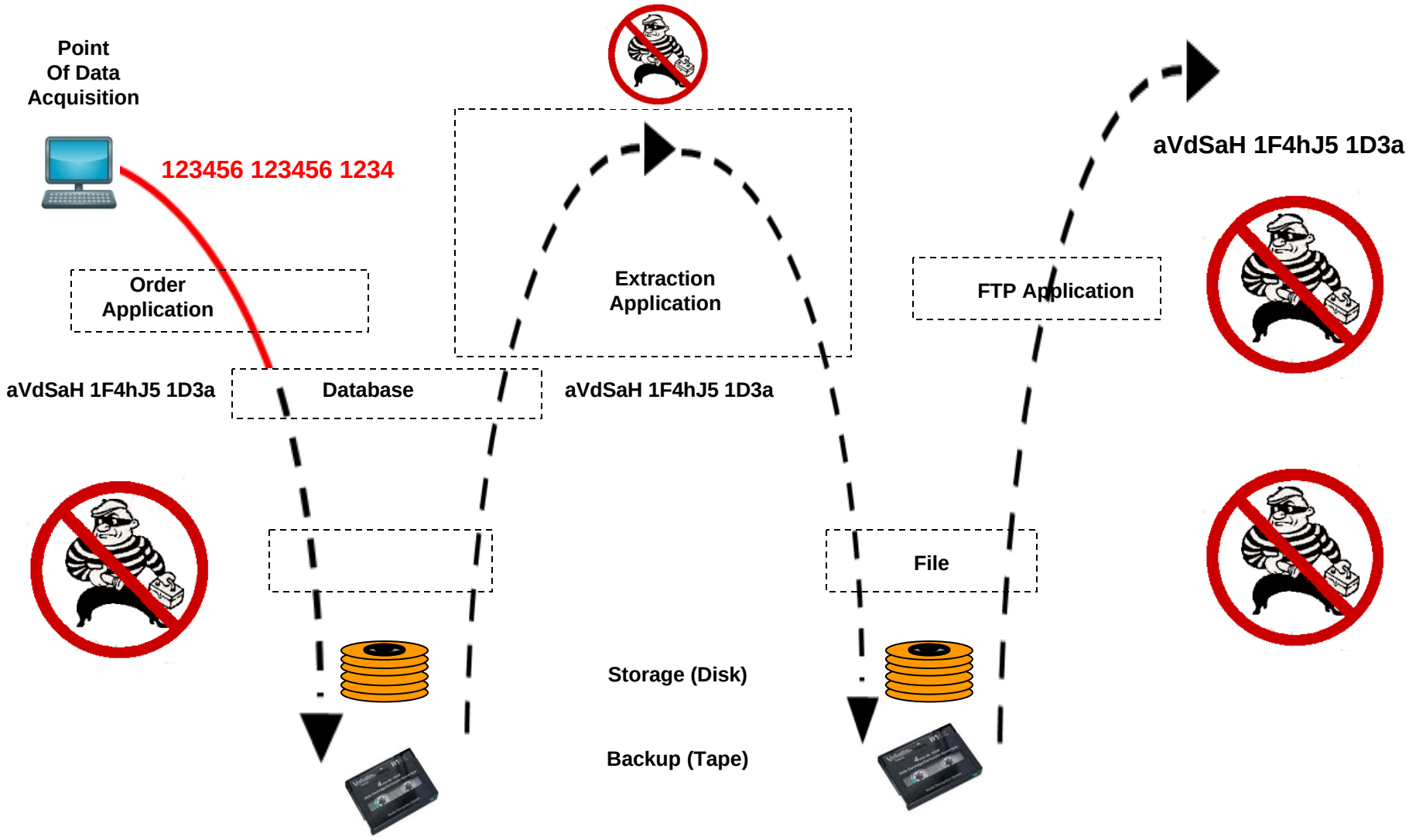


Case 2b: Goal – Addressing Advanced Attacks & PCI

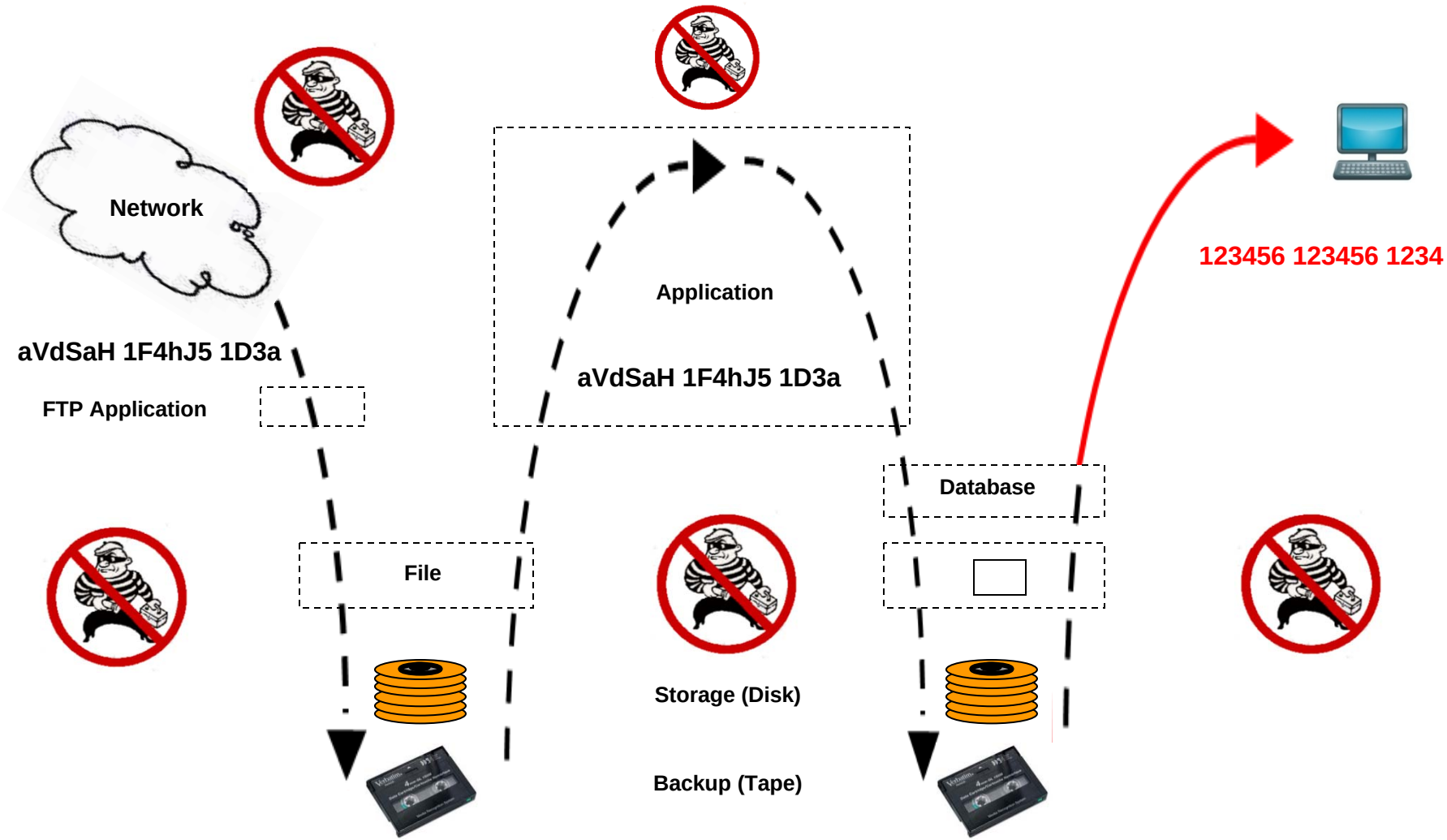


protegrity

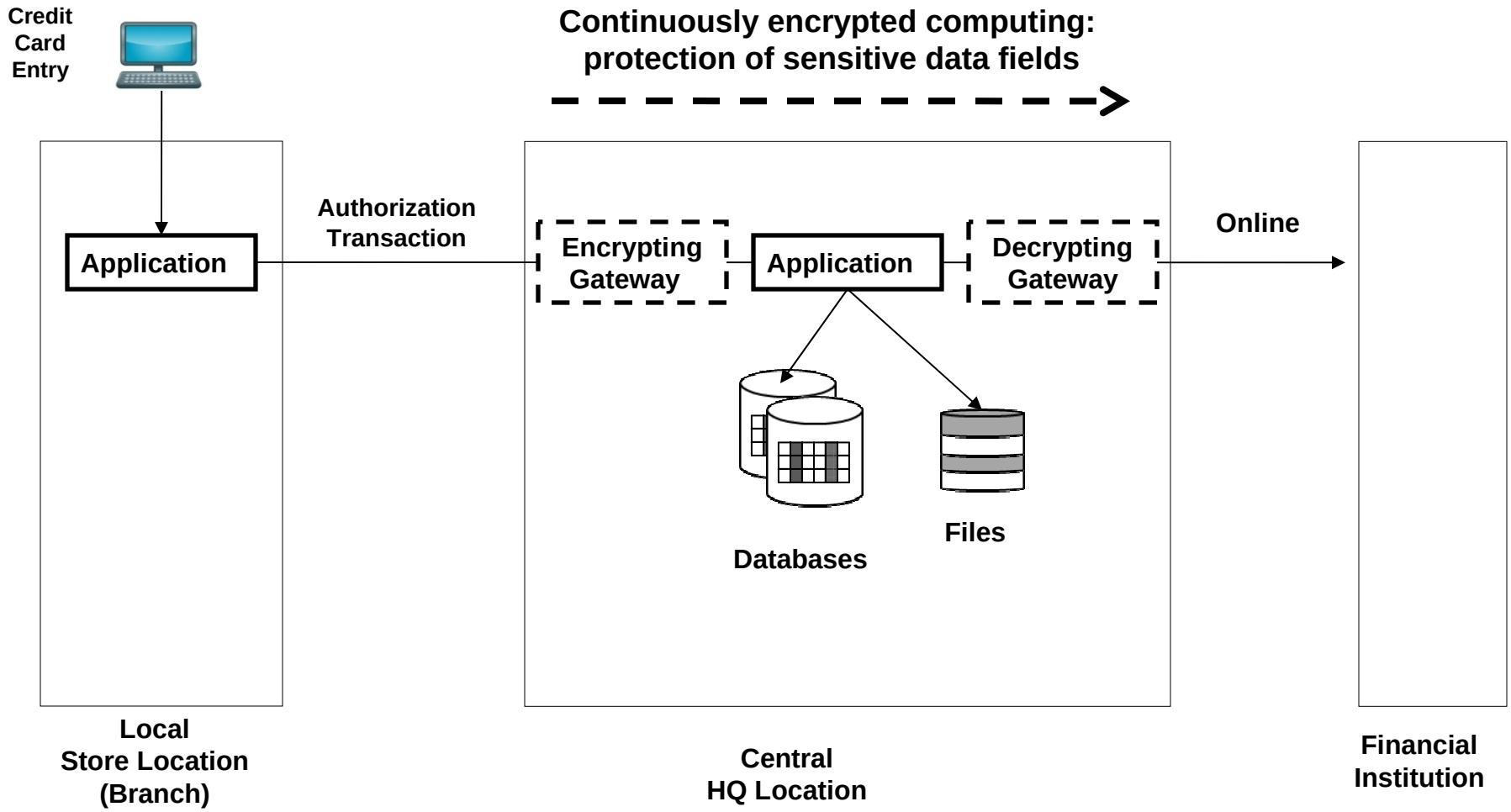
Case 2b: From Encrypted Database to File & FTP



Case 2b: From Selectively Encrypted File to Encrypted Database

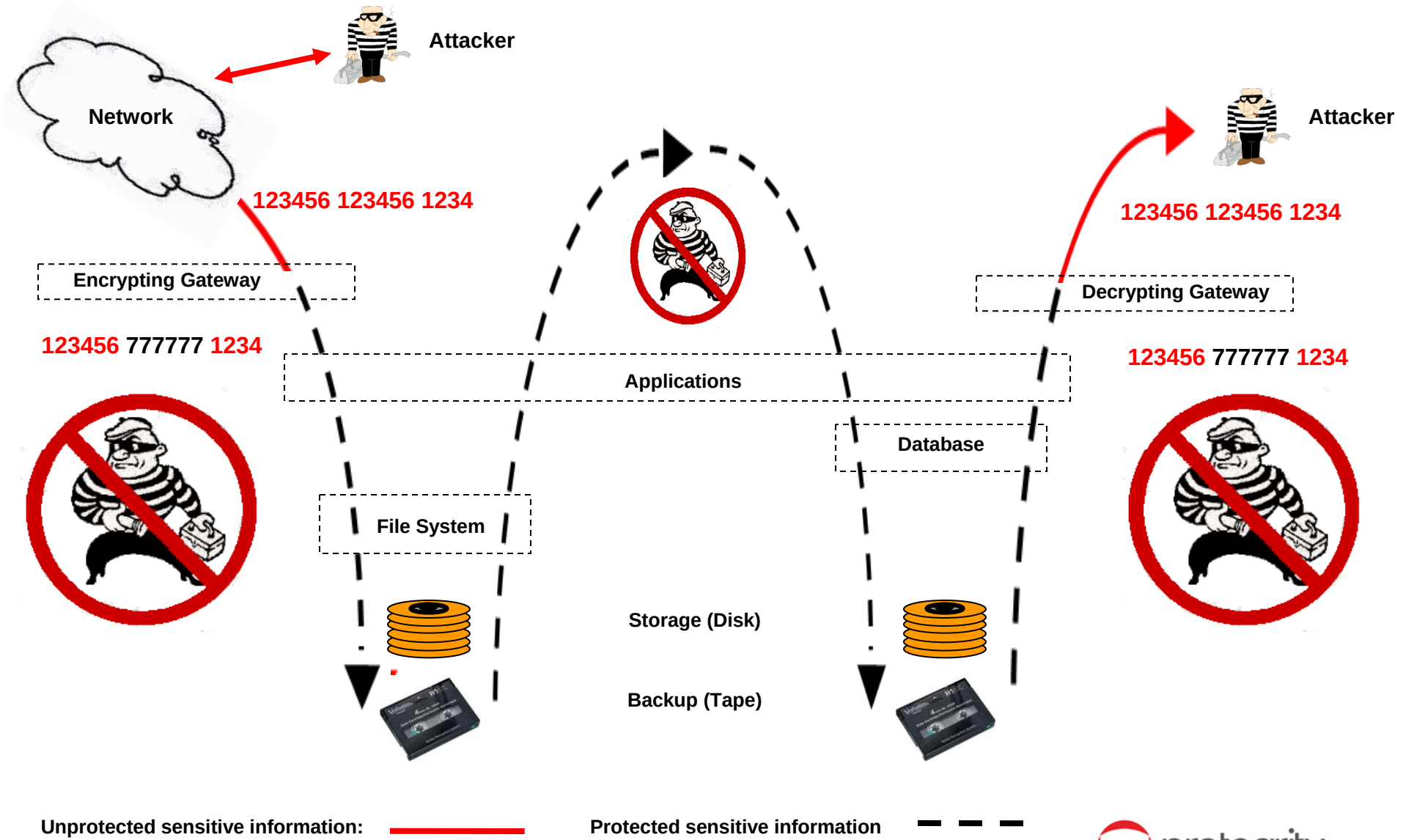


Case 3: Goal – Addressing Advanced Attacks & PCI



protegrity

Case 3: Gateway Encryption



http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1051481

http://ssrn.com/abstract=1126002

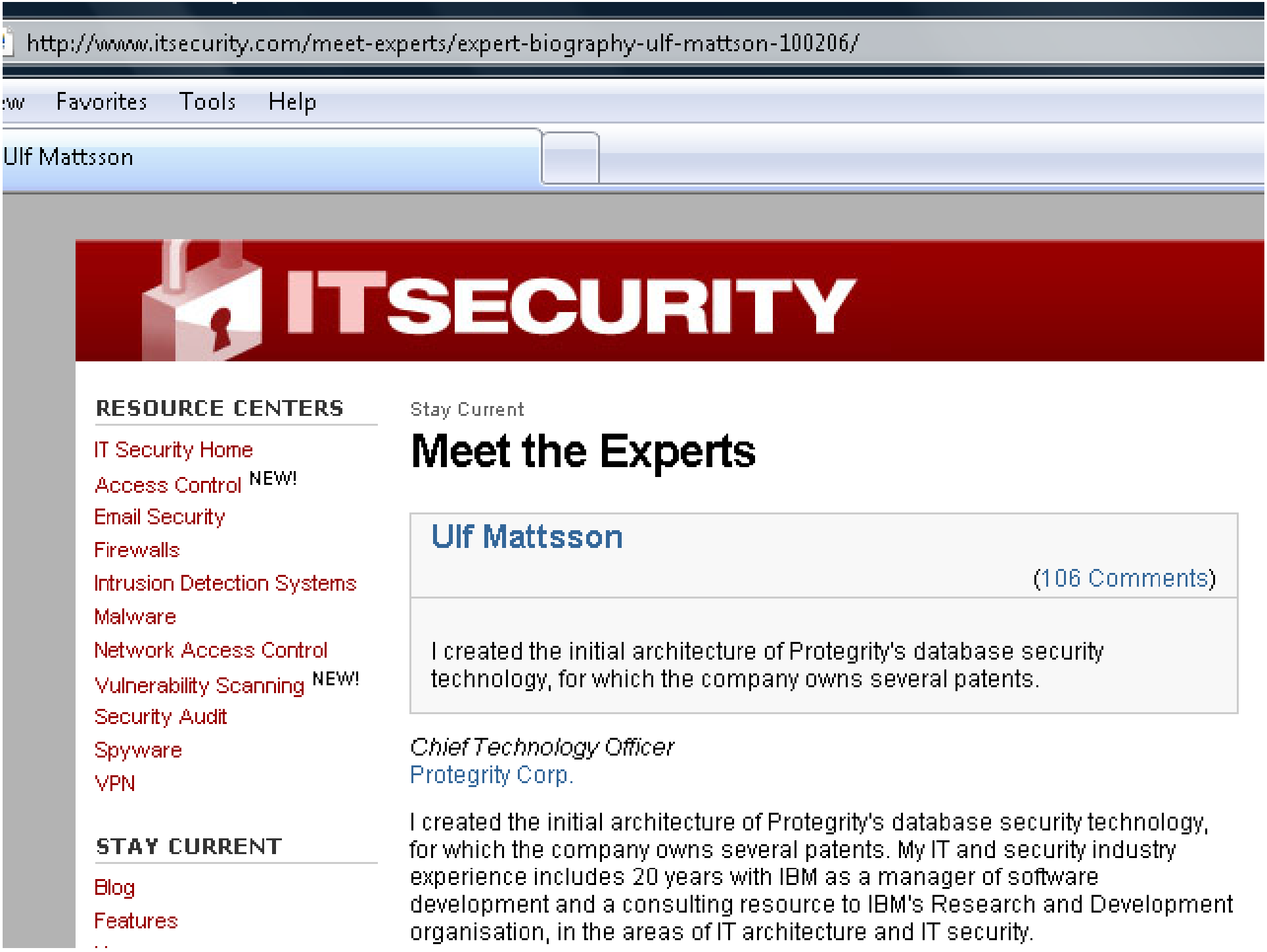
http://www.net-security.org/dl/insecure/INSECURE-Mag-2.pdf

http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1144290

How to keep sensitive data locked down across applications, databases, and files, including ETL data loading tools, FTP processes and EDI data transfers.

protegrity

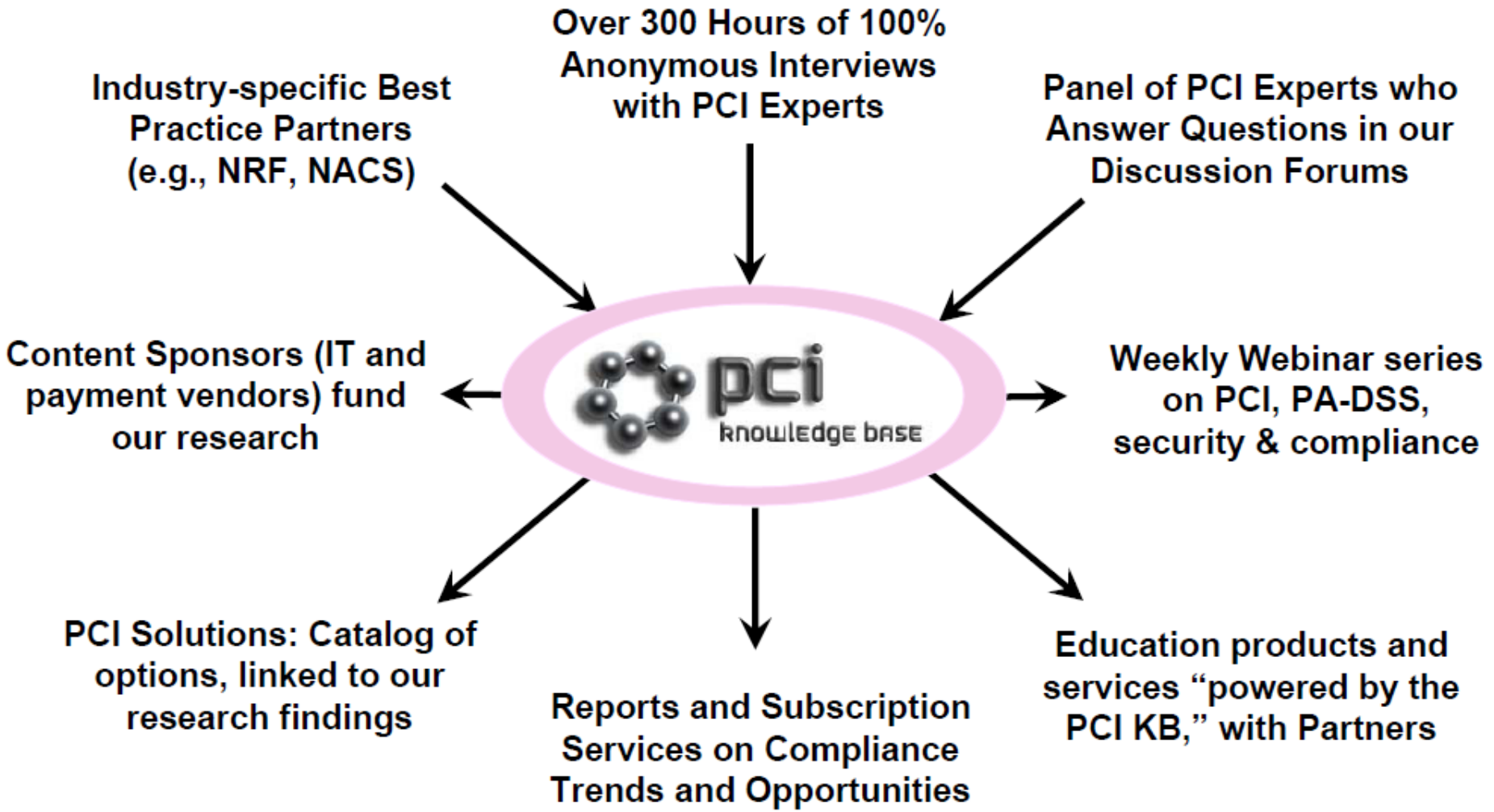
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=940287


http://www.itsecurity.com/meet-experts/expert-biography-ulf-mattson-100206/
Favorites
Tools
Help
Ulf Mattsson
ITSECURITY
RESOURCE CENTERS
IT Security Home
Access Control NEW!
Email Security
Firewalls
Intrusion Detection Systems
Malware
Network Access Control
Vulnerability Scanning NEW!
Security Audit
Spyware
VPN
STAY CURRENT
Blog
Features
Stay Current
Meet the Experts
Ulf Mattsson
(106 Comments)
I created the initial architecture of Protegrity's database security technology, for which the company owns several patents.
Chief Technology Officer
Protegrity Corp.
I created the initial architecture of Protegrity's database security technology, for which the company owns several patents. My IT and security industry experience includes 20 years with IBM as a manager of software development and a consulting resource to IBM's Research and Development organisation, in the areas of IT architecture and IT security.

Organizations are now required to protect sensitive data, or face the wrath of public consequences - be that public disclosure to your customers or regulatory non-compliance. With growing incidents of intrusions across industries and strong regulatory requirements to secure private data, enterprises need to make DBMS security a top priority.

http://www.quest-pipelines.com/newsletter-v7/0706_C.htm

What is The PCI Knowledge Base?



Source: PCI Knowledge Base, March 2009