

Design and Development of an Advanced Database System with Multitenant Architecture, Advanced Security Transparent Data Encryption & Data Redaction

Mohammed Waheeduddin Hussain¹, Prof. P. Premchand², Dr. G. Manoj Someswar³

¹Professor & Head, Department of CSE, Nawab Shah Alam Khan College of Engineering & Technology, Affiliated to JNTUH, Malakpet, Hyderabad – 500024, Telangana, India

²Professor in Department of Computer Science & Engineering, University College of Engineering, Osmania University. Hyderabad-500007, Telangana, India

³Professor & Dean (Research), Department of CSE, Nawab Shah Alam Khan College of Engineering & Technology, Affiliated to JNTUH, Malakpet, Hyderabad – 500024, Telangana, India

Abstract: *As organizations increase their reliance on possibly distributed information systems for daily business, they become more vulnerable to security breaches even as they gain productivity and efficiency advantages. Though a number of techniques, such as encryption and electronic signatures are currently available to protect data when transmitted across sites, a truly comprehensive approach for data protection must also include mechanisms for enforcing access control policies based on data contents, subject qualifications and characteristics, and other relevant contextual information, such as time. It is well understood today that the semantics of data must be taken into account in order to specify effective access control policies. Also, techniques for data integrity and availability specifically tailored to database systems must be adopted. In this respect, over the years, the database security community has developed a number of different techniques and approaches to assure data confidentiality, integrity, and availability. However, despite such advances, the database security area faces several new challenges. Factors such as the evolution of security concerns, the "disintermediation" of access to data, new computing paradigms and applications, such as grid-based computing and on-demand business, have introduced both new security requirements and new contexts in which to apply and possibly extend current approaches. In this research paper, we first survey the most relevant concepts underlying the notion of database security and make a sincere effort to design and develop an advanced database system with a highly advanced secure mechanism keeping in view of the latest advancements in database security.*

Keywords: Advanced Security Transparent Data Encryption, Data Redaction, Multitenant Architecture, Enterprise Manager Sensitive Data Discovery, Clustering, Replication, Recovery Manager

1. Introduction

In the present day scenario, ever increasing security threats, diversifying compliance requirements, consolidation and cloud computing technology are just a few of the reasons to establish the fact that data security has become critical. Almost a decade after the first U.S. breach notification law, the need for strong preventive controls continues to increase as access to data expands. Stolen client devices, including tablets and smart phones have the potential to easily expose sensitive information as users move beyond the laptop. Outsourcing, offshoring, corporate mergers, and nearly continuous organizational change create additional risks by making it easier for malicious insiders to obtain sensitive data and for outside hackers to gain access to servers using social engineering attacks. These growing trends are just one reason why centralized and efficient protection of sensitive data regardless of the applications being used is more important than ever.[1] Putting in place security measures that consistently protect sensitive data at the source is a critical control needed as stored data continues to proliferate and access to data expands beyond traditional boundaries. Protecting data requires a defense in depth, multi-layered approach that encompasses preventive,

detective, and administrative controls. Advanced Security option delivers two essential preventive controls covering encryption of data-at-rest and redaction of sensitive data displayed by applications.[2] These controls help protect sensitive data from being exposed directly from storage or through applications. Oracle Advanced Security Transparent Data Encryption (TDE) helps prevent attacks that attempt to bypass the database and read sensitive information from data files at the operating system level, from database backups, or from database exports. Oracle Advanced Security Data Redaction complements TDE by reducing the risk of unauthorized data exposure in applications, redacting sensitive data in query results before the data leaves the database. This research paper describes TDE and Data Redaction and explains how these valuable preventive controls can work together to help secure users' sensitive data.

2. Objectives of Research Study

1) Preventing Database Bypass with Encryption

Data-at-rest encryption is an important control for blocking unauthorized access to sensitive data using methods that

circumvent the database. Privileged operating system accounts are just one of the vehicles used by attackers as well as insiders to gain access to sensitive information directly in physical storage.

Advanced Security Transparent Data Encryption (TDE) stops attackers from bypassing the database and reading sensitive information from storage by encrypting data in the database layer. Applications and users authenticated to the database continue to have access to application data transparently, while unauthenticated users attempting to circumvent the database are denied access to clear text data. To understand this better, consider the fact that privileged operating system users can access database tablespace files and extract sensitive data using simple shell commands. In addition, consider the possibility of attacks that read sensitive data from lost, stolen, or improperly decommissioned disks or backups. Figure 1 shows an example of extracting customer credit card numbers directly from storage using the common Linux “strings” command and a search pattern.

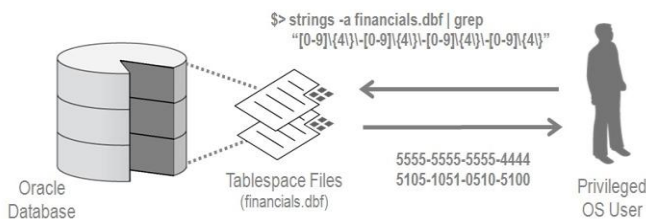


Figure 1: Extracting customer credit card numbers from Oracle database tablespace files

2) Advanced Security Transparent Data Encryption

Transparent Data Encryption resides at an optimal layer within the database to prevent database bypass while still being transparent to applications and easy to deploy. TDE can encrypt individual application table columns or entire application tablespaces. It is transparent to applications because the encryption and decryption processes do not require any application changes, and the application users do not have to directly deal with encrypted data.[3] Most importantly, TDE’s built-in two-tier encryption key management provides full key lifecycle management, tracking the keys across their lifetime with helpful metadata attributes, and assisted encryption key rotation, switching to a new master key with no downtime. Figure 2 shows how encrypting an Oracle database using TDE prevents database bypass.

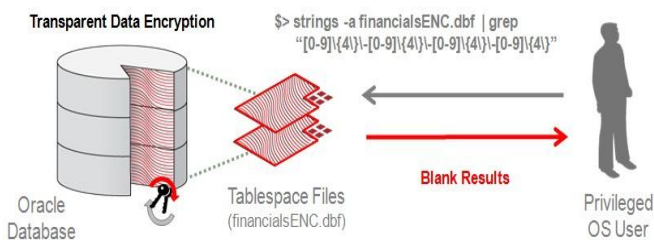


Figure 2: Encrypting with Transparent Data Encryption to prevent database bypass

TDE is unique when compared to other alternatives that encrypt entire storage volumes or require new toolkits and programming APIs. Alternative approaches do not protect

against many bypass attacks, may require significant application changes, have complex key management, and are not integrated with complementary technologies such as Advanced Compression, Recovery Manager and Multitenant.

The high level of protection provided by TDE follows common standards for strong encryption as described in the figure below.

Encryption Algorithms	Hashing Algorithms (optional)
Advanced Encryption Standard (AES) <ul style="list-style-type: none"> • Key length: 128, 192, 256 bits 	Secure Hash Algorithm 1 (SHA-1) <ul style="list-style-type: none"> • Digest length: 160 bits
Triple Data Encryption Standard <ul style="list-style-type: none"> • Key length: 168 bits 	

Figure 3: Standard encryption and hashing algorithms used by TDE

3) Protecting Sensitive Data Using TDE Column Encryption

Advanced Security TDE column encryption can be used to encrypt specific data in application tables such as credit card numbers and U.S. Social Security numbers. Customers identify columns within their application schema containing sensitive or regulated data, and then encrypt only those columns. This approach is useful when the database tables are large, only a small number of columns must be encrypted, and the columns are known. TDE column encryption also is useful for warehouse applications where each query is likely to return a very different set of data. Enterprise Manager Sensitive Data Discovery searches for and identifies sensitive columns quickly. Data encrypted using TDE column encryption remains encrypted on backup media and discarded disk drives, helping prevent unauthorized access and potential data breaches that bypass the database.

4) Protecting Entire Applications Using TDE Tablespace Encryption

Advanced Security TDE tablespace encryption protects entire application tables by encrypting the underlying tablespaces. It encrypts application tablespaces regardless of the data’s sensitivity an irrespective of its data type. Tablespace encryption simplifies the encryption process because there is no need to identify specific database columns. It is useful when the database contains a large amount of sensitive data to be encrypted and the columns reside in many different locations. TDE tablespace encryption and TDE column encryption can be used independently of one another or together within the same database. As with TDE column encryption, data encrypted with TDE tablespace encryption remains protected on backup media that could pose opportunities for bypass attacks.

3. Proposed System

3.1 Performance Characteristics

TDE’s cryptographic operations are extremely fast and well integrated with relational Database features. TDE leverages CPU-based hardware cryptographic acceleration available

in Intel® AES-NI and Oracle SPARC T4/T5 platforms to increase performance by up to 5x or more. The block-level operations of TDE tablespace encryption receive an additional performance boost from database buffering and caching. Tablespace encryption integrates seamlessly with Advanced Compression, ensuring that compression occurs before encryption. Tablespace encryption also integrates with the advanced technologies in Exadata such as Exadata Hybrid Columnar Compression (EHCC) and Smart Scans, which offload certain cryptographic processing to storage cells for fast parallel execution.

3.2 Built-In Key Management

Key management is critical to the security of the encryption solution. Advanced Security TDE provides an out-of-the-box, two-tier key management architecture consisting of data encryption keys and a master encryption key. The data encryption keys are managed automatically and are encrypted by the master encryption key. The master encryption key is stored and managed outside of the database within an Oracle Wallet (a standards-based PKCS12 file that protects keys). Keeping the master key separate from the encrypted data mitigates attacks because both the keys and the encrypted data must be separately compromised to gain access to clear data. The two-tier key architecture also enables rotation of master keys without having to re-encrypt all of the sensitive data. Advanced Security introduces a new dedicated SYSKM role for optional delegation to a designated user account that may run all key management operations including rotating master keys and changing the keystore password. Oracle Enterprise Manager provides a convenient graphical user interface for creating, rotating, and managing TDE master keys as shown in the figure below:

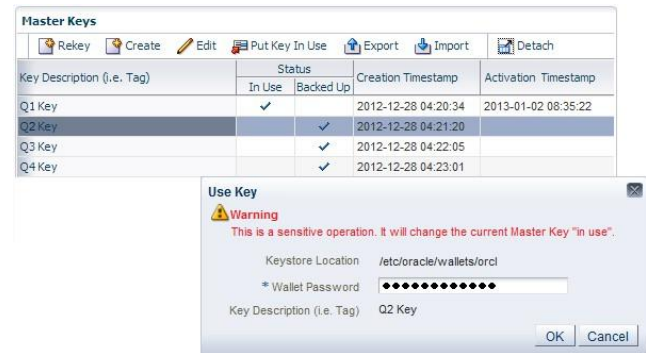


Figure 4: Managing and rotating TDE master keys using Relational Database Enterprise Manager

3.3 Encryption Impact for Common Operational Activities

Essential day-to-day database operational activities can potentially leak sensitive data when not performed properly, making bypass easy. Examples of these activities include database backup and restore, data movement, high-availability clustering, and replication.[4]

Advanced Security TDE supports these critical database operational activities and helps ensure that the data remains encrypted. Tablespace encryption integrates with Recovery Manager (backup and restore), Oracle Data Pump (data movement), Active Data Guard (redundancy and failover), and Golden Gate (replication). TDE also integrates with internal features of the database such as redo to prevent possible data leakage in logs.

Database Products and Technologies	Example Points of Integration	TDE Support
Exadata	Exadata Smart Scans, Exadata Hybrid Columnar Compression	✔
Database Compression	Oracle Advanced Compression	✔
Backup and Restore	Oracle Recovery Manager (RMAN), Oracle Secure Backup	✔
Export and Import	Oracle Data Pump Export and Import	✔
High-Availability Clusters	Oracle Real Application Clusters (RAC), Oracle Active Data Guard	✔
Storage Management	Oracle Automatic Storage Management (ASM)	✔
Database Replication	Oracle GoldenGate	✔

Figure 5: Example integrations with Advanced Security TDE

This fully integrated approach to database encryption makes the solution easy to deploy in complex real-world environments, while protecting against bypass attacks that attempt to take advantage of gaps in operational processes.

3.4 Limiting Sensitive Data Exposure with Data Redaction

Privacy and compliance require a cost effective approach to managing data exposure in applications. The embrace of smartphone and tablet devices make the issue of sensitive data exposure even more urgent as data access beyond the

traditional office environment becomes commonplace.[5] Even traditional applications require a more comprehensive solution for reducing exposure to sensitive data. Take for example, a call center application with a screen that exposes customer credit card information and personally identifiable information to call center operators. Today, exposure of that information, even to valid application users, may violate privacy regulations and put the data at unnecessary risk.



Figure 6: Clear and redacted information displayed in a call center application

3.5 Advanced Security Data Redaction

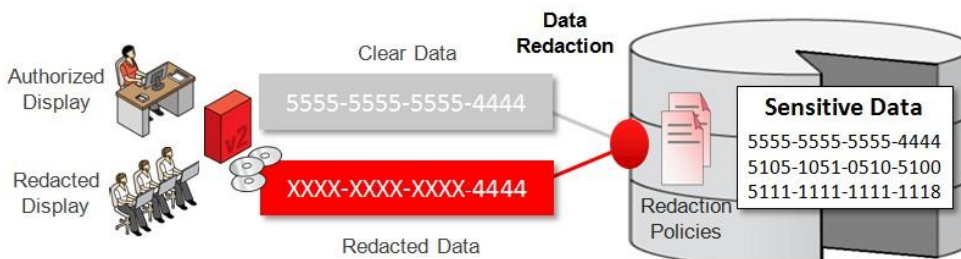


Figure 7: Redacting sensitive data displayed by applications using Data Redaction

4. Policies and Transformations

Advanced Security Data Redaction supports a number of different transformations that can redact all data in specified columns, preserve certain pieces of the data, or randomly generate replacement data. Examples of the supported data transformations are shown below:

	Stored Data	Redacted Display
Full	10/09/1992	01/01/2001
Partial	987-65-4328	XXX-XX-4328
RegExp	first.last@example.com	[hidden]@example.com
Random	5105105105105100	5500000000000004

Figure 8: Example Data Redaction transformation

Data Redaction makes the business need-to-know decision based on declarative policy conditions that utilize rich runtime contexts available from the database and from the applications themselves. Examples include user identifiers, user roles, and client IP addresses. Context information available from Oracle Application Express (APEX), Oracle Real Application Security, and Oracle Label Security also can be utilized. Redacting APEX applications is straightforward because you can create policy conditions using the application users and application identifiers that APEX automatically tracks. Multiple runtime conditions can be joined together within a Data Redaction policy for fine-grained control over when redaction occurs. The policies are stored and managed inside of the database, and they go into effect immediately upon being enabled.

Advanced Security Data Redaction provides selective, on-the-fly redaction of sensitive data in database query results prior to display by applications so that unauthorized users cannot view the sensitive data. The stored data remains unaltered, while displayed data is transformed and redacted on-the-fly before it leaves the database. Data Redaction reduces exposure of sensitive information and helps prevent exploitation of application flaws that may disclose sensitive data in application pages. It is well suited for both new and legacy applications that need to limit exposure of sensitive data without invasive application changes.

4.1 Performance Characteristics

High-speed performance is crucial for Data Redaction because the target databases typically will be production systems. Data needs to be transformed on-the-fly at runtime, without altering data stored on disk or in caches and buffers. Because the transformations will execute on production environments and will be repeated frequently, the performance overhead must be small. [6]

One important performance characteristic of Data Redaction is that it supports only data transformations with proven high performance. These are a subset of all the possible operations that could be used to transform data in non-production environments. This specific subset avoids long running and processor intensive operations.

Data Redaction also leverages performance optimizations of the Database that are only possible by being part of the database kernel. The implementation ensures that data transformations are fast in-memory computations. Policy information is cached in memory and policy expressions are evaluated only once per execution, so there is no per row performance impact.

4.2 Security Considerations

Another benefit from Data Redaction being part of the database kernel is tighter security. The implementation avoids tempering methods that plague other redaction techniques due to their dependence on proxies that can be meddled with. Additionally, Data Redaction in the kernel continues protecting when other security measures may be compromised. For example, runtime conditions in policies

can narrow the impact of a SQL Injection attack by continuing to redact sensitive data even when the attack has bypassed other preventive controls in the application and database.

Data Redaction also avoids obvious sources of leakage where the redaction policy could be bypassed by copying data into a new table that does not have a policy. Certain mass copy operations that touch redacted data are blocked by default, and this behavior can be overridden where necessary using a Data Redaction exempt privilege.

Although Data Redaction can be used to prevent accidental viewing of sensitive data by privileged database users such as DBAs, it is intended primarily for redacting data displayed by software applications. Data Redaction does not prevent privileged users from connecting directly to the database and running ad hoc queries that back into pieces of sensitive data (i.e. it does not stop exhaustive ad hoc queries or other inference attacks). However, note that Data Redaction is fully compatible with Oracle Database security solutions that control and monitor privileged database user access, including DBAs. It can be deployed in tandem with Oracle Database Vault or Oracle Audit Vault & Database Firewall. Data Redaction can be used with database encryption as well, and it is a great complement to TDE.

4.3 Easy to Deploy Data Redaction

Data Redaction can be deployed for existing applications quickly using either a command line API or Oracle Enterprise Manager. The command line API is a PL/SQL procedure that accepts protected columns, transformation types, and conditions. The Enterprise Manager provides a convenient Policy Expression Builder that enables administrators to define and apply redaction policies on existing applications. As shown below, the Policy Expression Builder dialog guides the user through creating policy conditions that use context obtained from applications, the database, the APEX framework, and other database security solutions.

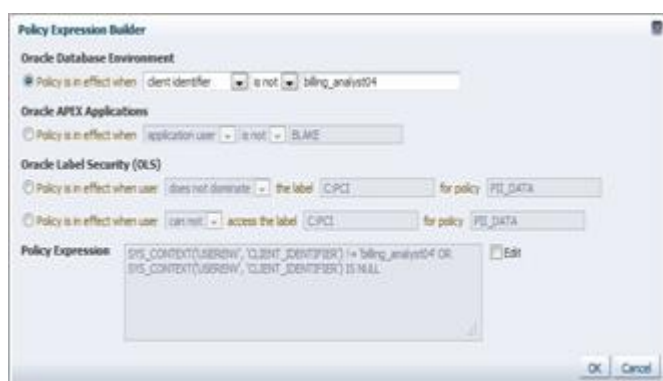


Figure 9: Using Enterprise Manager Policy Expression Builder to create Data Redaction policies

Predefined column templates also are available in Oracle Enterprise Manager for redacting common sensitive data such as credit card numbers and U.S. Social Security numbers. Enterprise Manager Sensitive Data Discovery assists in locating columns to be redacted inside of complex application schemas.

Another reason why Data Redaction is easy to deploy is its transparency to applications and the database. For application transparency, Data Redaction supports the column data types that are frequently used by applications and various database objects including tables, views, and materialized views. Redacted values retain key characteristics of the original data such as the data type and optional formatting characters. Random redaction values are drawn from data ranges defined by the existing column data. For transparency to the database, Data Redaction avoids impacting essential database operational activities. It does not affect administrative tasks such as data movement (Data Pump) or database backup and restore (Recovery Manager). It does not interfere with database cluster configurations such as Real Application Clusters, Active Data Guard, and GoldenGate. Data Redaction does not get in the way of existing database triggers or Virtual Private Database (VPD) policies. And because Data Redaction is part of the database kernel, no separate installation is required.

4.4 Comparison to Alternative Approaches

Traditional approaches to redacting sensitive data typically relied on application coding or installing third-party software on the database server to modify its behavior. These alternatives have important drawbacks compared to Data Redaction.

Approaches that require coding new application logic, modifying existing SQL statements, or authoring custom application scripts are likely to result in disparate solutions that are inconsistent across the enterprise and costly to maintain over their lifetime. In addition, strict controls must be placed on new application development to make sure that custom application code and new objects are properly accessed. The code also needs to take into consideration multiple factors under which the redaction policies are enforced, while maintaining the performance and semantics of the application. Approaches that add new components to the Oracle Database, overwrite existing components, establish proxies, and modify basic behavior of the database also are fraught with problems. Not only do the new components introduce new attack surfaces, but they also can create performance overhead, impact operational activities of the database, and may fail when attempting to transform complex database queries that are generated by applications.

Approaches that add new components to the Oracle Database, overwrite existing components, establish proxies, and modify basic behavior of the database also are fraught with

In contrast, redacting directly in the Oracle Database kernel using Data Redaction has tighter security, superior performance, and better compatibility with a range of database configurations, use cases, and workloads.

5. Results

By Applying Encryption and Redaction in Multitenant Architecture of the proposed Advanced Database, Advanced Security fully supports the proposed advanced multitenant architecture of the advanced Database. Both TDE and Data

Redaction attributes automatically follow Pluggable Databases (PDB) as they move between multitenant container databases. When moving a PDB that has redaction policies, the policies transfer directly to the new container as part of the PDB. When moving an encrypted PDB, the TDE master keys for that PDB are transferred separately from the encrypted data to maintain proper security separation during transit. Encryption and redaction immediately resume their normal operation after the PDB has been plugged in and configured.

6. Conclusion & Future Scope

As data exposed in applications continues to rapidly expand, enterprises must have strong controls in place to protect data no matter what devices or applications are used. This proposed database helps organizations keep their sensitive information safe in this increasingly complex environment by delivering preventive, detective and administrative controls that enforce data security in the database.

This proposed database system provides two critical preventive controls:

- 1) Transparent Data Encryption encrypts data at rest to stop database bypass attacks from accessing sensitive information in storage.
- 2) Data Redaction reduces exposure of sensitive information in applications by redacting database query results on-the-fly, according to defined policies.
- 3) Together these two controls form the foundation of a multi-layered, defense-in-depth approach. The research further established that this proposed Database will be the world's most advanced database security solution.

References

- [1] Jaquith, A. (2007). Security metrics: Replacing fear, uncertainty, and doubt. Redwood City, CA: AddisonWesley Professional.
- [2] Knox, D. C. (2004). Effective Oracle database 10g security by design. New York: McGraw-Hill/Osborne.
- [3] Phifer, L. (2010). Top ten data breaches and blunders of 2009. eSecurity Planet, February 10. Retrieved from <http://www.esecurityplanet.com/features/article.php/3863556/Top-Ten-Data-Breaches-andBlunders-of-2009.htm>
- [4] Ponemon, L. (2009). Fourth annual US cost of data breach study. Ponemon Institute sponsored by PGP Corporation. Retrieved January 31, 2010, from <http://www.ponemon.org/local/upload/fckjail/generalcontent/18/file/2008-2009%20US%20Cost%20of%20Data%20Breach%20Report%20Final.pdf>
- [5] Privacy Rights Clearing House. (2010). Chronology of data breaches. Retrieved January 31, 2010, from <http://www.privacyrights.org>.
- [6] Staddon, J. (2003). Dynamic inference control. Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA, USA, 94-100.

Author Profile



Mohammed Waheeduddin Hussain awarded Master of Computer Science and Applications (MCA) from Aligarh Muslim University, Aligarh, UP in 1992, M.Tech(IT) from Punjabi University Patiala, Punjab in 2004, MBA from Osmania University, Hyderabad in 2012. He is pursuing Ph.D in Computer Science and Engineering from Acharya Nagarjuna University(ANU), Guntur, Andhra Pradesh. He has 24 years of Experience in the teaching field. Presently he is working as an Associate Professor and HOD of CSE& IT in Nawab Shah Alam Khan College of Engineering and Technology, Hyderabad. His Research interest includes Cloud Computing, Advanced Database Systems. He had published papers in national and international journals, presented his papers in national and International conferences to enhance his Research.



Prof. P. Premchand presently working as a Professor in Department of Computer Science & Engg., University College of Engineering, Osmania University. Hyderabad-500007. Andhra Pradesh, India. Guided several candidates towards the Award of Ph.D degree from JNTU, Hyderabad and ANU, Guntur, A.P.



Dr. G. Manoj Someswar, B.Tech., M.S.(USA), M.C.A., Ph.D. is having 26+ years of relevant work experience in Academics, Teaching, Industry, Consultancy, Research and Software Development. At present, he is working as a Professor & Dean (Research), Department of CSE, Nawab Shah Alam Khan College of Engineering & Technology, Affiliated to JNTUH, Malakpet, Hyderabad – 500024, Telangana, India and utilizing his teaching skills, knowledge, experience and expertise to achieve the goals and objectives of the Engineering College in the fullest perspective. He has attended more than 100 national and international conferences, seminars and workshops. He has more than 70 publications to his credit both in national and international journals. He is also having to his credit more than 50 research articles and paper presentations which are accepted in national and international conference proceedings both in India and Abroad. He has guided 15 Ph.D. research scholars from different Universities in India. He received National Awards like Rajiv Gandhi Vidya Gold Medal Award for Excellence in the field of Education and Rashtriya Vidya Gaurav Gold Medal Award for Remarkable Achievements in the field of Education.