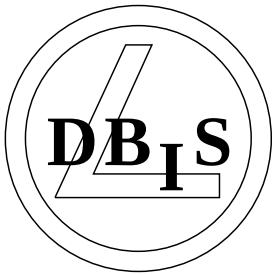
Workshop „Mobile Datenbanken und Informationssysteme:
Grundlagen, Technologien/Produkte, Anwendungen“

Verschlüsselung zur Erhöhung der Datensicherheit in mobilen Datenbanken



Thomas Fanghänel
thf@informatik.uni-jena.de

Friedrich-Schiller-Universität Jena
Institut für Informatik
Lehrstuhl für Datenbanken und Informationssysteme

Informatik 2002, Dortmund *2. Oktober 2002*

Gliederung

1. Motivation
2. Verschlüsselung im Schichtenmodell: Architekturdiskussion
3. Verschlüsselung in der Realität: Produktlösungen
4. Prototyp
5. Zusammenfassung und Ausblick

Motivation

Schutzziele:

- *Vertraulichkeit* — Nur autorisierte Nutzer dürfen auf Daten (lesend) zugreifen.
- *Integrität* — Nur autorisierte Nutzer dürfen Daten ändern, und zwar nur auf autorisierte Art und Weise.
- *Verbindlichkeit* — Informationen über Zugriffe und zugreifende Nutzer müssen verfügbar und Änderungen an Daten müssen autorisierten Nutzern zuordenbar sein.
- *Verfügbarkeit* — Autorisierte Nutzer können jederzeit auf die Daten zugreifen.

Schutzmaßnahmen:

- Müssen auf unterschiedlichen Ebenen stattfinden: Nutzer-, Geräte-, Betriebssystem-, Netzwerk- und DBMS-Ebene.
- Techniken: Nutzerauthentifizierung, Zugriffskontrollmechanismen, Verschlüsselung.
- Mögliche Schutzmaßnahmen unterscheiden sich für mobile und nicht-mobile Datenbanksysteme und Anwendungsumgebungen. Besonders: Geräteebene, Betriebssystemebene.

Implikationen

Zum Beispiel:

- Fehlendes Nutzerkonzept auf Betriebssystemebene.
  $\longrightarrow$ Nutzerverwaltung auf DBMS- oder Anwendungsebene

- Keine Authentifizierung oder Zugriffskontrolle für persistent gespeicherte Daten auf Ebene des Betriebssystems.
  $\longrightarrow$ Verschlüsselung und Authentifizierung auf DBMS- oder Anwendungsebene

Aber Vorsicht:

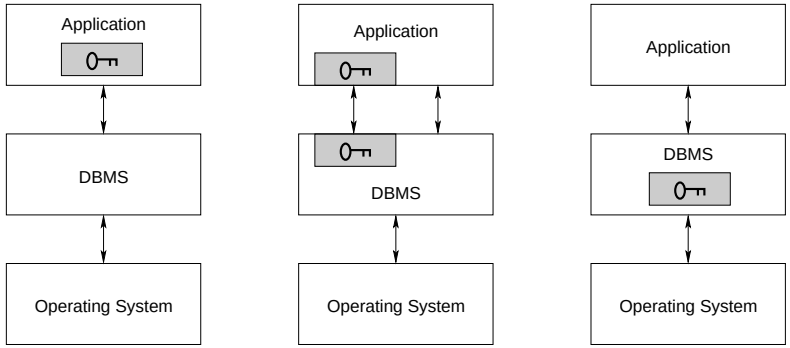
- Verschlüsselung aus anderer Motivation als bei zentralisierten Datenbanken!

- Nutzerkonzept nicht unbedingt notwendig, wenn man *single user devices* annimmt.

- Nutzerkonzept abwandelbar, kann z. B. auf Anwendungsebene bezogen werden.

Verschlüsselung im DBMS-Kontext

Auf welche Art läßt sich Verschlüsselung in einen DBMS-Kontext integrieren?



Welche prinzipiellen Probleme können dabei auftreten?

- Verschlüsselung einer Menge von Elementen zerstört Eigenschaften über dieser Menge (z. B. Ordnung, Vergleichbarkeit, Ähnlichkeit von Elementen).

- Im DBMS-Kontext wird das relevant, wenn solche Elementeigenschaften ausgenutzt werden (d. h. bei Verschlüsselung logischer DB-Granulate).

Anwendungsintegrierte Verschlüsselung

Anwendungsprogramm verschlüsselt Daten bevor diese an das DBMS übergeben werden.

Vorteile:

- Selektive Verschlüsselung möglich (z. B. einzelne Attributwerte, oder Teile davon).
- Anwendung bleibt unabhängig vom darunterliegenden DBMS.

Nachteile:

- Anwendungsabhängigkeit der verschlüsselten Daten.
- Stark eingeschränkter benutzbarer SQL-Sprachumfang für verschlüsselte Daten:
  - keine Bereichsanfragen
  - keine Prädikate
  - keine Integritätssicherung
  - keine Joins
- Keine Benutzbarkeit von Zugriffspfaden.

Die Anwendung muß auf wesentliche DBMS-Funktionalität verzichten, bzw. diese selbst implementieren.

DBMS-basierte Verschlüsselung

Anwendung nutzt DBMS-Erweiterungen zur Implementierung und Kapselung von Verschlüsselung (z. B. UDFs oder Trigger).

Vorteile:

- Kleines Verschlüsselungsgranulat möglich (z. B. Attribute oder Teile davon).
- Teilweise Anwendungsunabhängigkeit der verschlüsselten Daten.

Nachteile:

- Keine Integritätssicherung über verschlüsselten Daten durch das DBMS.
- Benutzbarkeit von Indexen sehr stark eingeschränkt (nur Punktanfragen).
- Verschlüsselungswissen steckt implizit in der Anwendung.

DBMS-basierte Verschlüsselung sichert immer noch keine Anwendungsunabhängigkeit der Daten, und erlaubt nur eingeschränkte Nutzbarkeit von DB-Funktionalität.

Bemerkung: Ansatz hat durchaus praktische Relevanz! (IBM DB2 UDB, Oracle 9*i*)

DBMS-integrierte Verschlüsselung

Interessantester Fall: Verschlüsselung ist vollständig ins Datenbanksystem integriert, d. h. in eine seiner Schichten.

Verschlüsselung auf einem für die jeweilige Schicht spezifischen Datenbankgranulat.

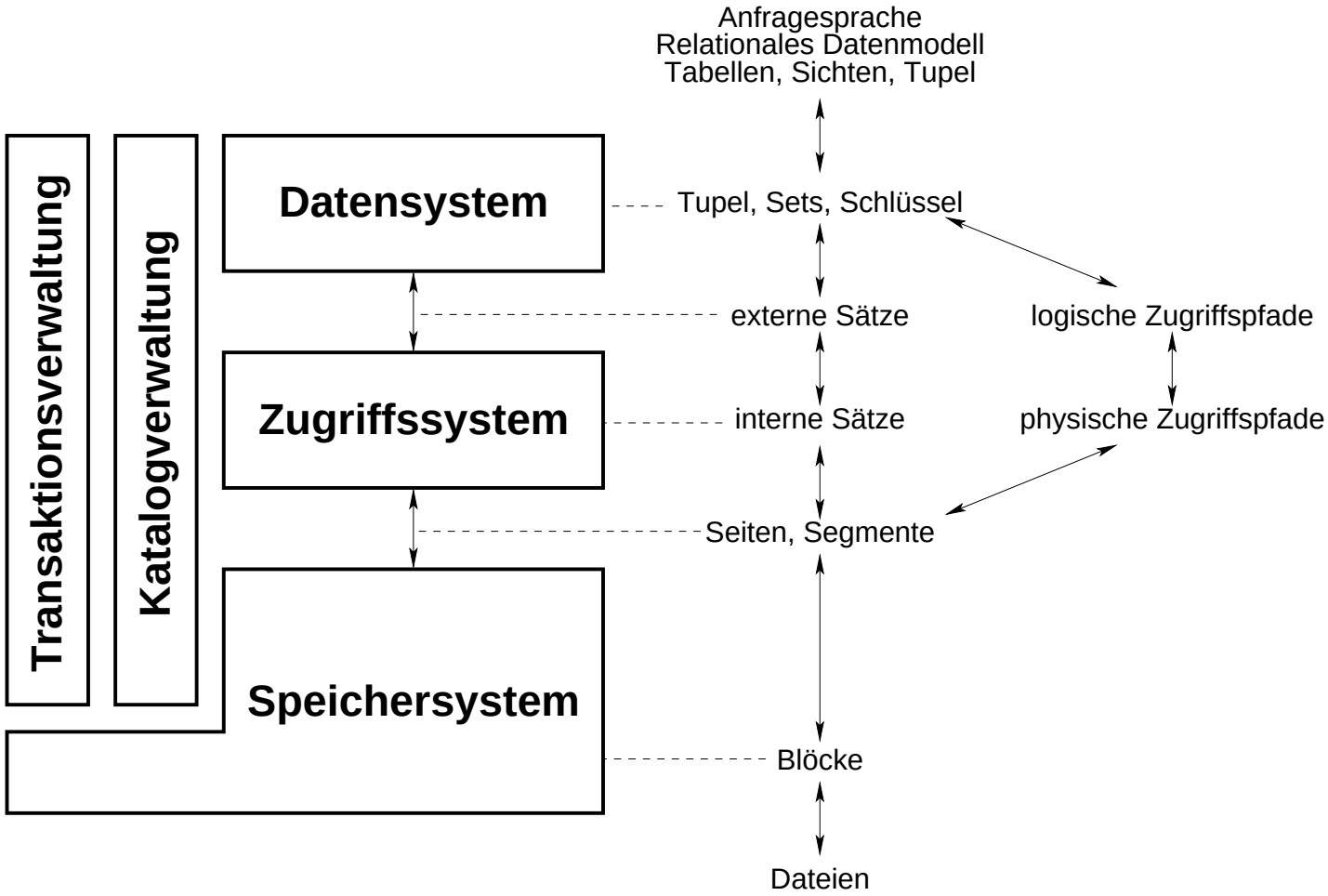
Vorteile:

- Anwendungsunabhängige verschlüsselte Datenablage (d. h. keine DML-Erweiterungen).
- Volle DB-Funktionalität nutzbar.

Nachteil:

- Anwendung nicht DBMS-unabhängig, da Verschlüsselung nicht standardisiert.

Schichtenmodell eines datenunabhängigen DBMS



DBMS-integrierte Verschlüsselung

Verschlüsselung im Datensystem:

- Logisches Verschlüsselungsgranulat (Attribute, Tupel, Tabellen, ...), ähnlich zu DBMS-basiertem Ansatz.
- Integritätssicherung durch Datenbanksystem möglich.
- Interpreter und Zugriffspfade benutzen verschlüsselte Repräsentationen.

Verschlüsselung im Zugriffssystem:

- Logisches Verschlüsselungsgranulat (externe oder interne Sätze).
- Trennung von Daten und Zugriffspfaden, damit z. B. Indizierung von unverschlüsselten Repräsentationen.

DBMS-integrierte Verschlüsselung

Verschlüsselung im Speichersystem:

- Verschlüsselung von Seiten, Segmenten oder Blöcken, d. h. physische Granulate.
  $\longrightarrow$ Implikationen in Bezug auf die Größe des externen Verschlüsselungsgranulates.
- Kein Unterschied zwischen Nutzerdaten, Katalogdaten, Indexstrukturen, . . .
- Pufferung von unverschlüsselten Daten im Seitenpuffer.

Für das Speichersystem sind verschlüsselte Dateien lediglich charakterisiert durch aufwendigen/teuren Sekundärspeicherzugriff.
Damit generische Behandlung durch Optimierer/Interpreter des DBMS.

Produktüberblick I

Oracle 9i Lite:

- Verschlüsselung auf Datenbankebene, externe Dienstprogramme für Ver- und Entschlüsselung.
  $\longrightarrow$ Speichersystem-integrierter Ansatz, basiert auf physischen Granulaten.
- Schutz durch DB-Passwort, kein Nutzerkonzept.
- Änderung des Datenbankpassworts erfordert Neuverschlüsselung der Datenbank.

iAnywhere Adaptive Server Anywhere 8.0:

- Verschlüsselung auf Datenbankebene, inklusive Logs, Katalogen und Zugriffspfaden. Interface in DDL integriert.
  $\longrightarrow$ Speichersystem-integrierter Ansatz, nutzt physische Granulate.
- Kein Nutzerkonzept, globales DB-Passwort.
- Änderung des DB-Passworts durch Entschlüsselung und Neuverschlüsselung der Datenbank.

Produktüberblick II

Microsoft SQL Server CE Edition 1.1:

- Verschlüsselung auf Datenbankebene, Nutzerinterface auf DDL-Level.
  $\longrightarrow$ Speichersystem-integrierter Ansatz
- Kein Nutzerkonzept, lediglich globales DB-Passwort.
- Passwortänderung durch Neuverschlüsselung der gesamten Datenbank.

IBM DB2 Everyplace Release 8.1:

- Verschlüsselung auf Tabellenebene, Interface in DDL integriert.
- Lösung ist ins Speichersystem integriert, und verschlüsselt Blöcke.
- Nutzerkonzept vorhanden, Privilegien auf Datenbankebene.
- Globaler Schlüssel für die gesamte Datenbank, allerdings nicht von einem Passwort abhängig.
- Passwortänderungen eines Nutzers erfordern nicht Neuverschlüsselung der Datenbank.

Prototyp auf Basis von DB2 Everyplace

Wunschliste:

- Verschlüsselung und Privilegierung auf Tabellenebene.
- Verschlüsselung nutzt einen Schlüssel pro Tabelle, und entkoppelt Passwörter und Nutzerdaten.

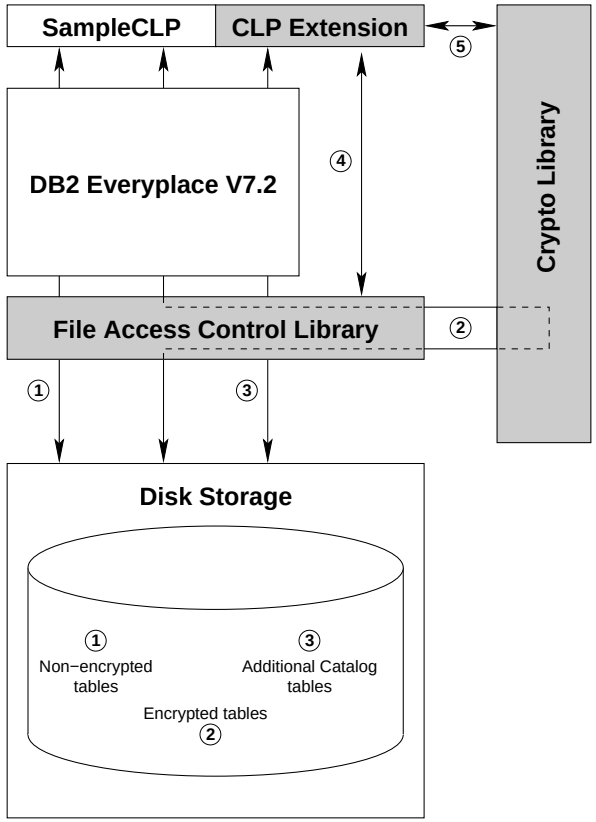
Nutzerinterface:

- Nur interaktiv benutzbar über erweiterten Command Line Processor (CLP).
- Nutzerverwaltung und Verschlüsselungsfunktionalität über Erweiterungen des SQL-Sprachschatzes (implementiert im CLP).
  - Nutzerverwaltung mittels `CREATE/DROP/ALTER USER`.
  - Verschlüsselung mittels `CREATE TABLE ... WITH ENCRYPTION`
  - Schlüsseländerungen mittels `REENCRYPT TABLE/DATABASE`

Prototyp verschmilzt Verschlüsselung und Nutzerkonzept weitgehend.

Aufbau des Prototypen

Prototyp besteht aus drei Teilen, welche den eigentlichen DB-Kern umschließen.



Zusammenfassung und Ausblick

Zusammenfassung:

- Datenverschlüsselung verfolgt im Kontext mobiler Datenbanken andere Ziele als im Umfeld zentraler Datenbanken.
- Integration ins DBMS lohnt sich für mobile Datenbanken/Anwendungen.
- Integration ins Speichersystem erlaubt generische Sicht auf verschlüsselte Sekundärspeicher, inklusive einfache Einbindung in bestehende Optimizer/Interpreter-Infrastrukturen.

Ausblick:

(1) Noch immer kein Produkt mit adäquater Verschlüsselungskomponente. ;-)

(2) Verschlüsselung auf Tabellenebene sichert hohe Anwendungsperformance, und damit gute Benutzbarkeit.

(3) Optimierer muss auf verschlüsselte Tabellen angepaßt werden.

Fragen/Diskussion

Vielen Dank!