

Datenbanken und Datensicherheit

Prof. Dr. rer. nat. Peter Gerwinski

11. Januar 2024

Datenbanken und Datensicherheit

<https://gitlab.cvh-server.de/pgerwinski/dbs>

1 Einführung

2 Kurzeinführung Unix

3 Kurzeinführung TCP/IP

4 Relationale Datenbanken

...

4.9 Indizierung

4.10 Funktionen und Trigger

4.11 GUI-Zugriff

4.12 SQL Injection

4.13 Datensicherheit bei Datenbanken

4.14 Sonstige Datenbanken

5 Kryptographie

5.1 Einführung

...

...



Änderungen
vorbehalten

4 Relationale Datenbanken

4.10 Funktionen und Trigger

Funktionen:

- **PROCEDURE** entspricht einer **void**-Funktion in C.
- <https://www.postgresql.org/docs/15/sql-createprocedure.html>

Trigger:

- <https://www.sqltutorial.org/sql-triggers/>
- <https://www.postgresqltutorial.com/postgresql-triggers/creating-first-trigger-postgresql/>

4 Relationale Datenbanken

4.11 GUI-Zugriff

- Anwendung nutzt DBMS-Client-Bibliothek
GUI-Programmierung: wie gewohnt
- Spezialfall: Web-Anwendung

Beispiel: Programmiersprache PHP

- Integration in HTML-Quelltext: `<?php ... ?>`
- Objekt zur Kommunikation mit Datenbanken: PDO

Literatur:

- <https://www.postgresqltutorial.com/postgresql-php/connect/>
- <https://www.phptutorial.net/php-pdo/pdo-connecting-to-postgresql/>

4 Relationale Datenbanken

4.12 SQL Injection

Problem:

- Ein böswilliger Benutzer gibt über eine Benutzerschnittstelle (z. B. ein Web-Interface) Daten ein (z. B. einen „Namen“), die Sonderzeichen enthalten, damit sie als SQL-Befehle ausgeführt werden.
- Literatur: <https://xkcd.com/327/>

Lösung: Die Benutzerschnittstelle prüft die Daten auf Sonderzeichen und ersetzt diese durch geeignete Escape-Sequenzen

- ' durch ' ' ersetzen
- Funktion `CHR ()`
- Viele DBMS verstehen ein vorangestelltes \.

Bessere Lösung: *Prepared Statements*

- <https://www.postgresql.org/docs/current/sql-prepare.html>
- https://www.w3schools.com/php/php_mysql_prepared_statements.asp

4 Relationale Datenbanken

4.13 Datensicherheit bei Datenbanken

- kein direkter Zugriff von außen auf die Datenbank
- feingranulare Benutzerrechte
- Software aktuell halten
- Prepared Statements
- Transportverschlüsselung

4 Relationale Datenbanken

4.14 Sonstige Datenbanken

- Eingebettete Datenbanken:
Berkeley DB, SQLite
Software-Bibliothek, keine Client-Server-Struktur
- Nicht-relationale Datenbanken:
dokumentenorientierte Datenbanken, noSQL
Performanz wichtiger als Konsistenz
→ Applikationen stärker in Konsistenzprüfung eingebunden

5 Kryptographie

5.1 Einführung

Was ist Datensicherheit?

(CIA)

- Vertraulichkeit (confidentiality) → Verschlüsselung
- Integrität (integrity) → Konsistenzprüfungen, Prüfwerte, Signaturen
- Verfügbarkeit (availability) → Backups, Ausfallsicherheit

- Identifizierbarkeit (Authentizität, Nichtabstreitbarkeit, Zurechenbarkeit)
→ Passwörter, Signaturen
bzw.
- Anonymität (plausible Abstreitbarkeit, Nichtzurechenbarkeit)
→ Pseudonymisierung, Anonymisierung,
Verschlüsselung, Steganographie

5 Kryptographie

5.1 Einführung

Was ist Datensicherheit?

- (CIA)
- Vertraulichkeit (confidentiality) → Verschlüsselung
- Integrität (integrity) → Konsistenzprüfungen, Prüfwerte, Signaturen
- Verfügbarkeit (availability) → Backups, Ausfallsicherheit
- Identifizierbarkeit (Authentizität, Nichtabstreitbarkeit, Zurechenbarkeit)
→ Passwörter, Signaturen
bzw.
- Anonymität (plausible Abstreitbarkeit, Nichtzurechenbarkeit)
→ Pseudonymisierung, Anonymisierung,
Verschlüsselung, Steganographie → Kryptographie

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: Cäsar-Chiffre, Monoalphabetische Substitution

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: Cäsar-Chiffre, Monoalphabetische Substitution, One Time Pad

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: Cäsar-Chiffre, Monoalphabetische Substitution, One Time Pad

- Problem: Schlüsselaustausch

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: Cäsar-Chiffre, Monoalphabetische Substitution, One Time Pad, Pseudozufallszahlengenerator, Startwert als Schlüssel
- Problem: Schlüsselaustausch

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: Cäsar-Chiffre, Monoalphabetische Substitution, One Time Pad, *spezielle* Pseudozufallszahlengeneratoren, Startwert als Schlüssel: Enigma, DES, 3DES, RC4, IDEA, Blowfish, TwoFish, CAST, AES, ...
- Problem: Schlüsselaustausch

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: **Cäsar-Chiffre**, **Monoalphabetische Substitution**,
One Time Pad,
spezielle Pseudozufallszahlengeneratoren, Startwert als Schlüssel:
Enigma, **DES**, **3DES**, **RC4**, IDEA, Blowfish, TwoFish, CAST, AES, ...
- Problem: Schlüsselaustausch

5 Kryptographie

Kryptographie

- Verschlüsselung: symmetrisch, asymmetrisch, hybrid
- Hashes: Einwegfunktionen, Salt
- Signaturen, Zertifikate
- Schlüsselaustausch

5.2 Symmetrische Verschlüsselung

- Derselbe Schlüssel zum Ver- und Entschlüsseln
- Beispiele: **Cäsar-Chiffre**, **Monoalphabetische Substitution**, One Time Pad, *spezielle* Pseudozufallszahlengeneratoren, Startwert als Schlüssel: **Enigma**, **DES**, **3DES**, **RC4**, IDEA, Blowfish, TwoFish, CAST, AES, ...
- Problem: Schlüsselaustausch
- Lösung: *asymmetrische Verschlüsselung*

5.3 Asymmetrische Verschlüsselung

- verschiedene Schlüssel zum Ver- und Entschlüsseln:
öffentlicher und privater Schlüssel
- Prinzip: mathematische Operation,
einfach durchzuführen, schwer rückgängig zu machen
- Beispiel: $N = p \cdot q$ – einfacher als Primfaktorzerlegung von $N \longrightarrow$ RSA
 $73 \cdot 97 = 7081$: geht notfalls noch im Kopf
Primfaktorzerlegung von 7081: mindestens schriftlich, besser mit Rechner
- Beispiel: $c = b^a$ – einfacher als $a = \log_b c \longrightarrow$ Diffie-Hellman, ElGamal
 $7^5 = 16807$: geht notfalls noch im Kopf
 $\log_7 16807$: mindestens schriftlich, besser mit Rechner

→ Details: Algorithmen und Datenstrukturen

5.3 Asymmetrische Verschlüsselung

- verschiedene Schlüssel zum Ver- und Entschlüsseln:
öffentlicher und privater Schlüssel
- Prinzip: mathematische Operation,
einfach durchzuführen, schwer rückgängig zu machen
- Beispiel: $N = p \cdot q$ – einfacher als Primfaktorzerlegung von $N \longrightarrow$ RSA
 $73 \cdot 97 = 7081$: geht notfalls noch im Kopf
Primfaktorzerlegung von 7081: mindestens schriftlich, besser mit Rechner
- Beispiel: $c = b^a$ – einfacher als $a = \log_b c \longrightarrow$ Diffie-Hellman, ElGamal
 $7^5 = 16807$: geht notfalls noch im Kopf
 $\log_7 16807$: mindestens schriftlich, besser mit Rechner

→ Details: Algorithmen und Datenstrukturen

- Nachteil: wesentlich aufwendiger und daher langsamer
als symmetrische Verschlüsselung

→ *hybride Verschlüsselung*: nur Schlüsselaustausch asymmetrisch,
eigentliche Verschlüsselung symmetrisch

5.4 Signaturen

- *kryptographische Hash-Funktion*: leicht auszurechnen, schwer zu manipulieren
- asymmetrisch: *Signatur*
Hash-Wert mit privatem Schlüssel verschlüsseln,
mit öffentlichem Schlüssel entschlüsseln
- symmetrisch: *Message Authentication Code* (MAC)
z. B. Hash-Wert über Nachricht + geheimer Schlüssel

5.4 Signaturen

- *kryptographische Hash-Funktion*: leicht auszurechnen, schwer zu manipulieren
- asymmetrisch: *Signatur*
Hash-Wert mit privatem Schlüssel verschlüsseln,
mit öffentlichem Schlüssel entschlüsseln
- symmetrisch: *Message Authentication Code* (MAC)
z. B. Hash-Wert über Nachricht + geheimer Schlüssel

Angriffsmöglichkeit: *Man-in-the-middle*-Angriff

Beim Schlüsselaustausch anderen Schlüssel unterschieben

→ Sorgfalt beim Schlüsselaustausch

5.4 Signaturen

- *kryptographische Hash-Funktion*: leicht auszurechnen, schwer zu manipulieren
- asymmetrisch: *Signatur*
Hash-Wert mit privatem Schlüssel verschlüsseln,
mit öffentlichem Schlüssel entschlüsseln
- symmetrisch: *Message Authentication Code* (MAC)
z. B. Hash-Wert über Nachricht + geheimer Schlüssel

Angriffsmöglichkeit: *Man-in-the-middle*-Angriff

Beim Schlüsselaustausch anderen Schlüssel unterschieben

→ Sorgfalt beim Schlüsselaustausch

Praxis-Beispiele

- SSH
- HTTPS
- OpenPGP