Last time

- Internet Application Security and Privacy
  - Application-layer security and privacy: remailers, PGP/gpg, OTR
This time

- Finish OTR

- Database Security
  - Introduction to Databases
  - Security Requirements
  - Integrity
  - Auditability, Access Control, and Availability
Deniable Authentication

• Do **not** want digital signatures
  - Non-repudiation is great for signing contracts, but undesirable for private conversations

• But we **do** want authentication
  - We can't maintain privacy if attackers can impersonate our friends

• Use Message Authentication Codes
  - We talked about these earlier
No Third-Party Proofs

• Shared-key authentication
  – Alice and Bob have the same MK
  – MK is required to compute the MAC
  – How is Bob assured that Alice sent the message?

• Bob cannot prove that Alice generated the MAC
  – He could have done it, too
  – Anyone who can verify can also forge

• This gives Alice a measure of deniability
Using these techniques

Using these techniques, we can make our online conversations more like face-to-face “off-the-record” conversations.

But there's a wrinkle:
- These techniques require the parties to communicate interactively.
- This makes them unsuitable for email.
- But they're still great for instant messaging!
Off-the-Record Messaging

- Off-the-Record Messaging (OTR) is software that allows you to have private conversations over instant messaging, providing:
  - Encryption
    - Only Bob can read the messages Alice sends him
  - Authentication
    - Bob is assured the messages came from Alice
Off-the-Record Messaging

- Perfect Forward Secrecy
  - Shortly after Bob receives the message, it becomes unreadable to anyone, anywhere

- Deniability
  - Although Bob is assured that the message came from Alice, he can't convince Charlie of that fact
  - Also, Charlie can create forged transcripts of conversations that are every bit as accurate as the real thing
Off-the-Record Messaging

• Availability of OTR:
  - It's built in to Adium X (a popular IM client for OSX)
  - It's a plugin for pidgin (a popular IM client for Windows, Linux, and others)
    • With these two methods, OTR works over almost any IM network (AIM, ICQ, Yahoo, MSN, etc.)
  - It's a proxy for other Windows or OSX AIM clients
    • Trillian, iChat, etc.
  - Third parties have written plugins for other clients
    • Miranda, Trillian, Kopete
(Relational) Databases

- Structured, queryable collection of data (records)
- Each record consists of fields (elements)
- Structure (schema) set by database administrator
- Database management system (DBMS) provides support for queries and management
- Most popular DBMS is based on relational model
- Stores records in one or multiple tables (relations)
  - Table has named columns (attributes) and rows (tuples)
  - Individual tables can have relationships between them
<table>
<thead>
<tr>
<th>Name</th>
<th>First</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Airport</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAMS</td>
<td>Charles</td>
<td>212 Market St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>ADAMS</td>
<td>Edward</td>
<td>212 Market St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>BENCHLY</td>
<td>Zeke</td>
<td>501 Union St.</td>
<td>Chicago</td>
<td>IL</td>
<td>60603</td>
<td>ORD</td>
</tr>
<tr>
<td>CARTER</td>
<td>Marlene</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>CARTER</td>
<td>Beth</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>CARTER</td>
<td>Ben</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>CARTER</td>
<td>Lisabeth</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
<tr>
<td>CARTER</td>
<td>Mary</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
<td>CMH</td>
</tr>
</tbody>
</table>
Relations

<table>
<thead>
<tr>
<th>ADAMS</th>
<th>212 Market St.</th>
<th>Columbus</th>
<th>OH</th>
<th>43210</th>
</tr>
</thead>
<tbody>
<tr>
<td>BENCHLY</td>
<td>501 Union St.</td>
<td>Chicago</td>
<td>IL</td>
<td>60603</td>
</tr>
<tr>
<td>CARTER</td>
<td>411 Elm St.</td>
<td>Columbus</td>
<td>OH</td>
<td>43210</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charles</td>
</tr>
<tr>
<td>Edward</td>
</tr>
<tr>
<td>Zeke</td>
</tr>
<tr>
<td>Marlene</td>
</tr>
<tr>
<td>Beth</td>
</tr>
<tr>
<td>Ben</td>
</tr>
<tr>
<td>Lisabeth</td>
</tr>
<tr>
<td>Mary</td>
</tr>
</tbody>
</table>

| CMH | 43210 | 60603 | ORD |

19-11
Database Queries

• Most popular query language is SQL
  - `SELECT First FROM NAME-ZIP
    WHERE (Zip = '43210') AND (Name = 'ADAMS')`
    • Prints first names of people in relation NAME-ZIP whose zip code is 43210 and whose last name is Adams
  - `SELECT Name, Airport
    FROM NAME-ZIP, ZIP-AIRPORT
    WHERE NAME-ZIP.Zip = ZIP-AIRPORT.Zip`
    • Prints each person’s last name and his/her airport by joining relations NAME-ZIP and ZIP-AIRPORT
  - `SELECT COUNT(Name) FROM NAME-ZIP
    WHERE City = 'Chicago'
    • Prints number of families in Chicago
    • Can also do other computations, like `SUM`, `MIN`, or `AVG`
• Result of a query is a subschema
Security Requirements

- Physical database integrity
- Logical database integrity
- Element integrity
- Referential integrity
- Auditability
- Access control
- User authentication
- Availability
Database Integrity

- Protects against database corruption
- Allow only authorized individuals to perform updates
- Recover from physical problems
  - Power failures, disk crashes,....
- Perform periodic backups
- Keep log of transactions to replay transactions since last backup
Element Integrity

- Ensures correctness/accuracy of database elements
- **Access control** to limit who can update element
- **Element checks** to validate correctness
  - Element must be numeric, within a particular range,…
  - Not more than one employee can be president
  - Helps against mistakes by authorized users
  - Typically enforced by triggers (procedures that are automatically executed after an `INSERT, DELETE,`…)

19-15
Element Integrity (cont.)

- **Change log or shadow fields** to undo erroneous changes
  - In case the above fail, require additional space
- **Error detection codes** to protect against OS or hard disk problems
Integrity: Two-Phase Update

- For a set of operations, either all of them or none of them should be performed
  - Integrity violation if only some are performed

- First phase: gather information required for changes, but don’t perform any updates, repeat if problem arises

- Second phase: make changes permanent, repeat if problem arises

- See text for example
Integrity: Concurrency Control

- Concurrent modifications can lead to integrity violation
  - Two operations A and B read variable X
  - A then writes new value of X
  - B then writes new value of X
  - A’s update gets lost

- Need to perform A and B as **atomic** operations

- Take CS 454 for more about this
Referential Integrity

• Each table has a primary key
• Minimal set of attributes that uniquely identifies each tuple
  – User ID or social insurance number
  – First name and last name (maybe not)
• A table might also have a or multiple foreign keys, which are primary keys in some other table
  – Zip is (likely) a primary key in ZIP-AIRPORT
  – Zip is a foreign key in NAME-ZIP
• Referential integrity ensures that there are no dangling foreign keys
  – For each zip in NAME-ZIP, there is an entry in ZIP-AIRPORT
Auditability

• Keep an audit log of all database accesses
  – Both read and write

• Access control can be difficult (see later), audit log allows to retroactively identify users who accessed forbidden data
  – Police officer looking at somebody’s criminal record as a favor to a friend, unauthorized medical personnel looking at George’s Clooney’s medical record

• Maybe combination of accesses resulted in disclosure, not a single one (see later)

• Must decide about granularity of logging
  – Should results of a query be logged?
Access Control

- More difficult than OS access control
- Might have to control access at the relation, record or even element level
- Many types of operations, not just read/write
  - SELECT, INSERT, UPDATE, CREATE, DROP, ...
- Relationships between database objects make it possible to learn sensitive information without directly accessing it
  - Inference problem (see later)
- Efficiency problem in presence of thousands of records, each consisting of dozens of elements
Access Control (cont.)

- Access control might consider past queries
  - Current query, together with past ones, could reveal sensitive information
    - Iteratively querying whether element is in set ultimately leaks set
- Or type of query
  - `SELECT lastname, salary FROM staff WHERE salary > 50000` might be forbidden, but not
  - `SELECT lastname FROM staff WHERE salary > 50000`
User Authentication / Availability

- Database might do its own authentication
- Additional checks possible
  - E.g., time of day

- Databases facilitate sharing, but availability can suffer if multiple users want to access the same record
  - Block access until other user finishes updating record
Types of Data Disclosure

- Exact data
- Bounds
  - Sensitive value is smaller than H, but bigger than L
  - Might iteratively decrease range (binary search)
- Negative result
  - Knowing that a person does not have zero felony convictions is sensitive, even if actual number is hidden
- Existence
  - Knowing of existence of some data can be sensitive
- Probable value
  - Sensitive data has value x with probability y
Recap

- Finish OTR

- Database Security
  - Introduction to Databases
  - Security Requirements
  - Integrity
  - Auditability, Access Control, and Availability
Next time

- Database Security
  - Data Inference
  - Statistical Inference
  - Controls against Inference
- Multilevel Security Databases
  - Separation
  - Integrity Locks
  - Designs of MLS Databases
- Data Mining
  - Integrity and Availability
  - Privacy and Data Mining
  - Privacy-Preserving Data Mining