Module 3 Day 10

Authentication and Encryption

What makes an application?

- Program Data
  - ✓ Variables & .NET Data Types
  - ✓ Arrays
  - ✓ More Collections (list, dictionary, stack, queue)
  - ✓ Classes and objects (OOP)

- Program Logic
  - ✓ Statements and expressions
  - ✓ Conditional logic (if)
  - ✓ Repeating logic (for, foreach, do, while)
  - ✓ Methods (functions / procedures)
  - ✓ Classes and objects (OOP)
  - ❖ Frameworks (MVC)

- Input / Output
  - User
    - ✓ Console read / write
    - ✓ HTML / CSS
    - ❑ Front-end frameworks (HTML / CSS / JavaScript)
  - Storage
    - ✓ File I/O
    - ✓ Relational database
    - ❑ APIs

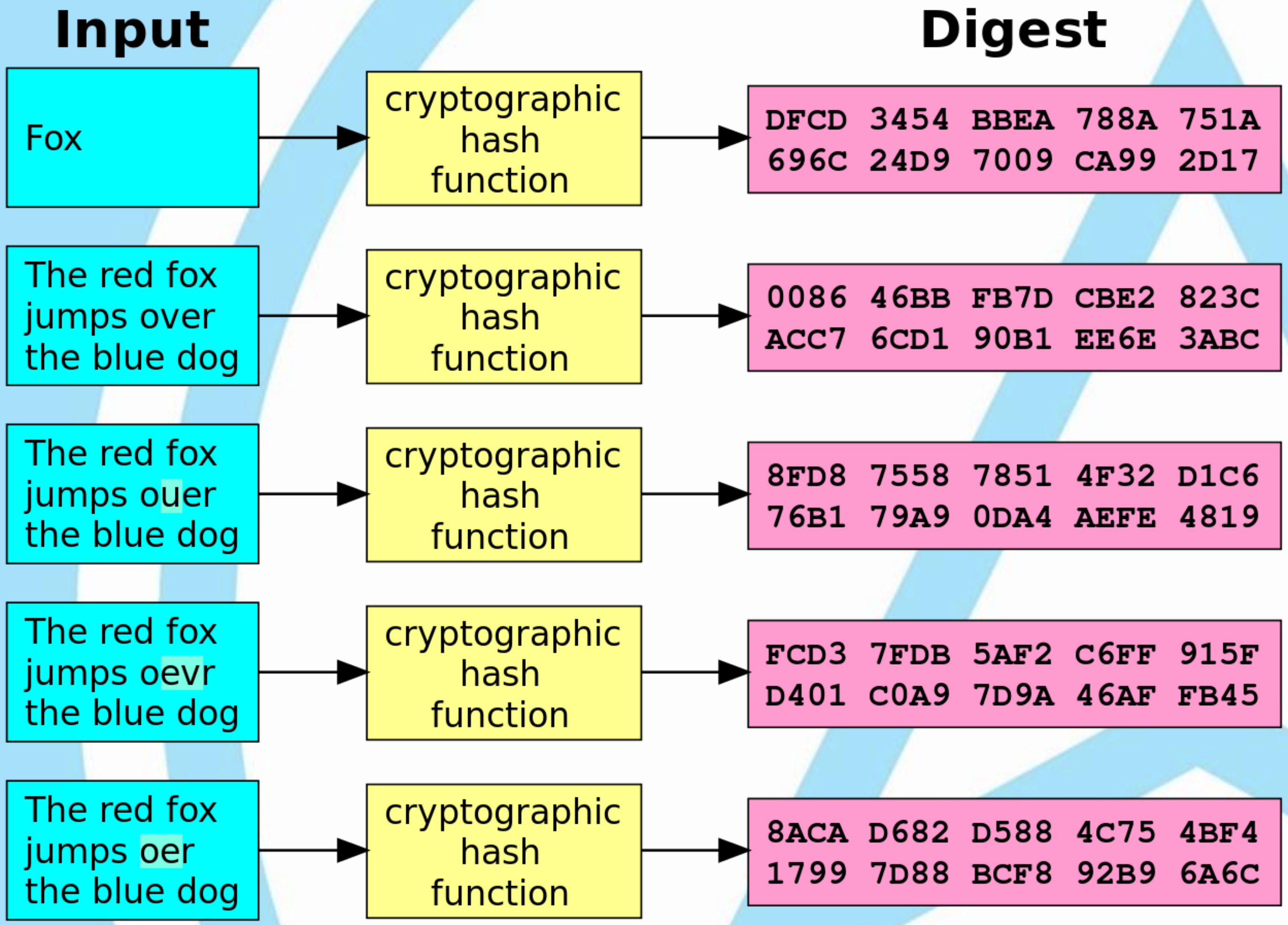
Authentication and Authorization

- Authentication
  - Who you are
  - Prove that you are who you claim to be
  - “What you have and what you know” (2FA)
    - ATM card, cell phone, ID card, biometric (fingerprint, retina, face)
    - Password, PIN, Security Question
  - Often involves a password
- Authorization
  - What you can do
  - Can you view data, or edit it? Delete it? Add users? Etc.
  - Useless without Authentication

Hashing Data

- One-way, repeatable algorithm to change data into a “hash value”
- One-way means there is no way to get to the original data, given only the hash
- Repeatable means if I run the same original data through the algorithm again, I’ll get the same result
- Used to verify data transmissions (aka, checksum)
- Used for storing passwords securely

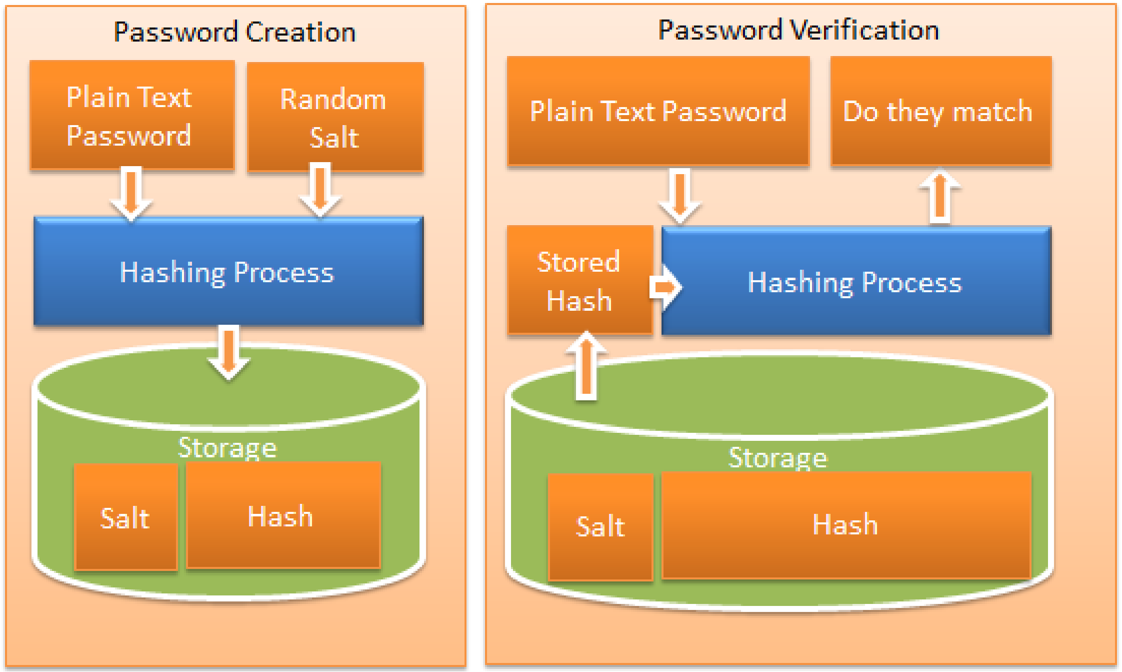
Hashing Data



Input		Digest
Fox	cryptographic hash function	DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog	cryptographic hash function	0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog	cryptographic hash function	8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog	cryptographic hash function	FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog	cryptographic hash function	8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

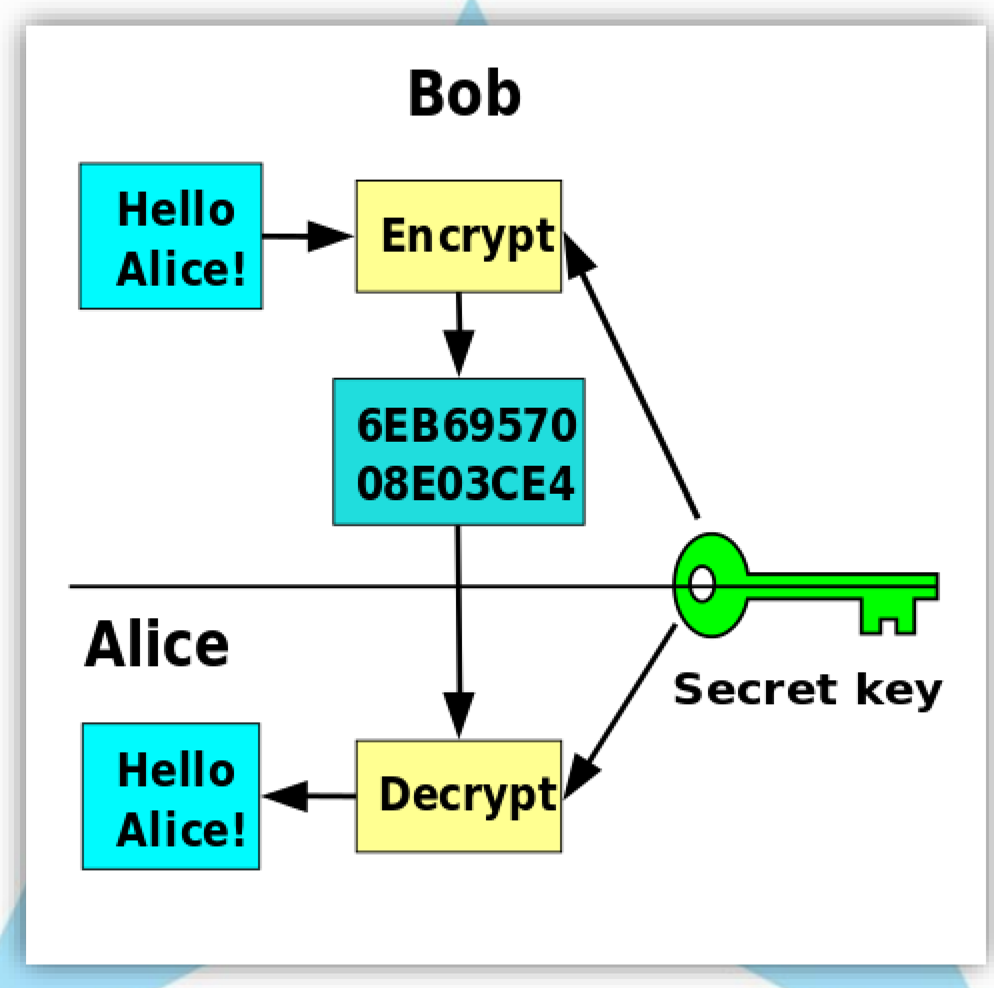
Hashing Passwords

- Password is hashed when created
  - Hash is stored in DB
- To login, password is hashed using the same algorithm
  - Hashes are compared.
- Adding a salt prevents dictionary attacks
  - Salt also stored in the DB
- Increasing work factor greatly increases security
  - Hash the hash



Encryption – Symmetric Key

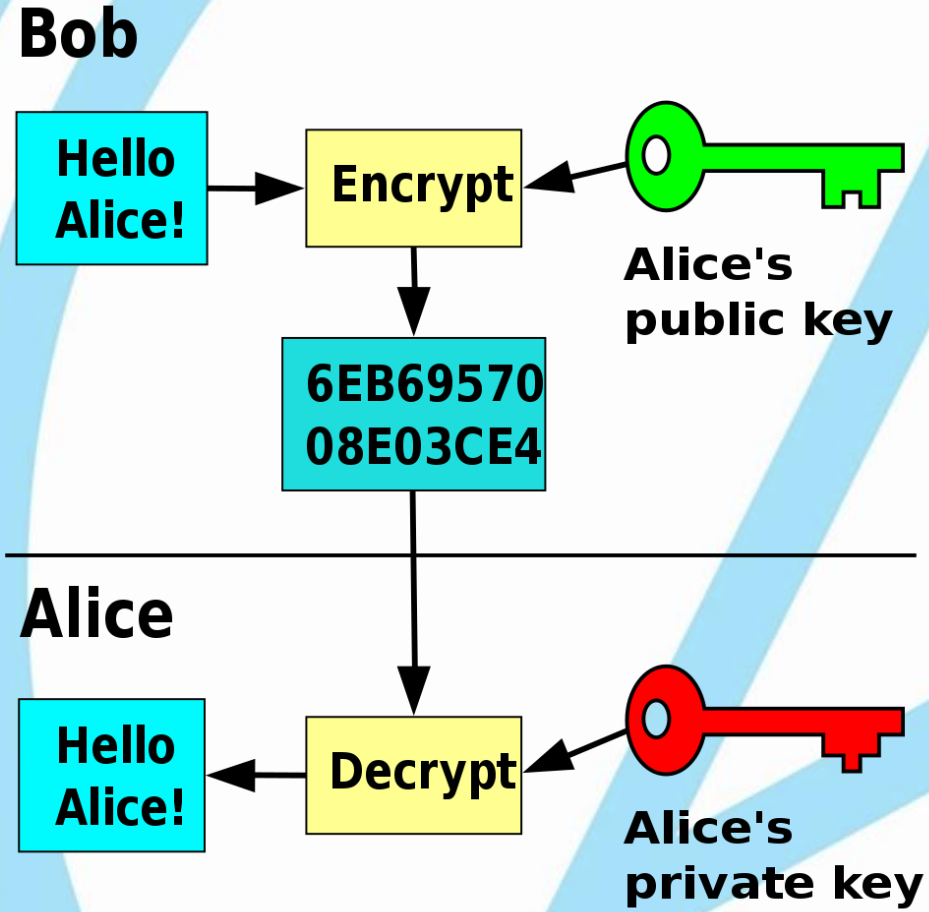
- Uses a single key to encrypt (lock) and decrypt (unlock) the data
- “Shared secret”
- Examples:
  - Password-protected files
  - Windows BitLocker



Encryption – Asymmetric Key

- Public key cryptography / Public Key Infrastructure (PKI)
- Two keys used: a “public” key and a “private” key
  - Messages encrypted using Public must be decrypted using Private
  - Message encrypted using Private must be decrypted using Public
- Can be used to
  - Securely send data to another user, or (encrypt public, decrypt private)
  - Guarantee the identity of the sender (encrypt private, decrypt public)

Bob securely sends message to Alice



Bob

Hello Alice!

Encrypt

Alice's public key

6EB69570 08E03CE4

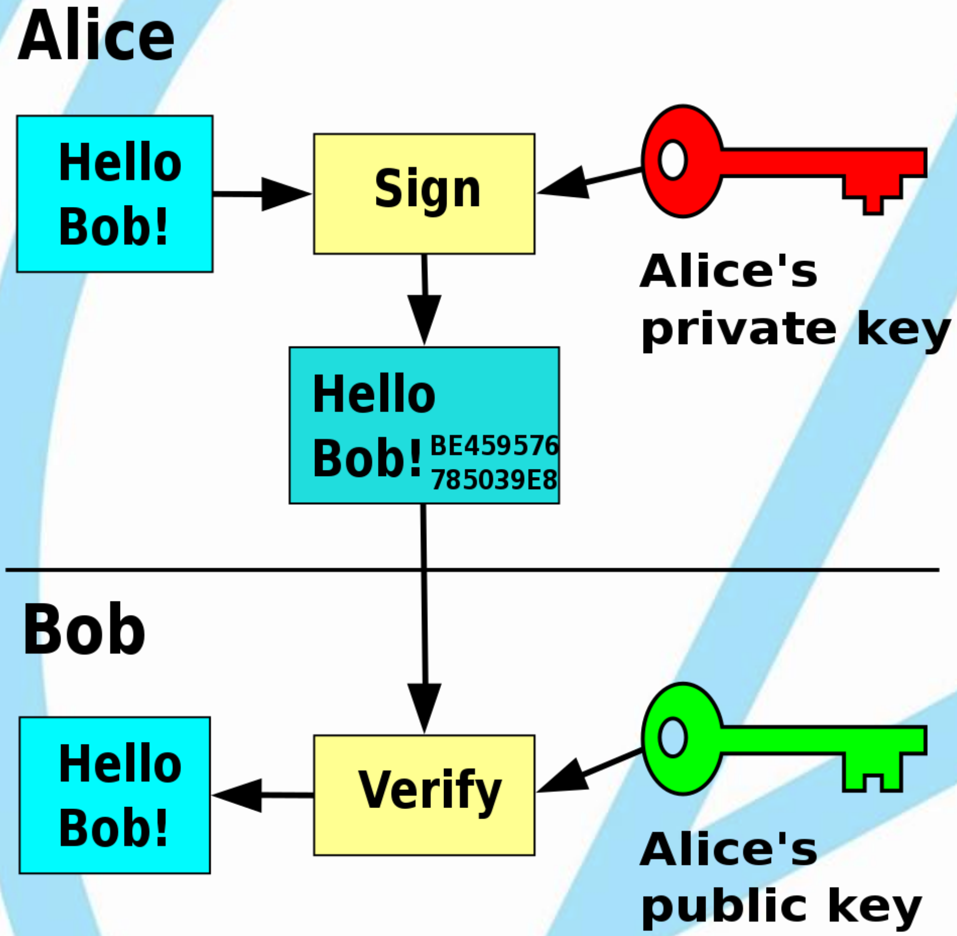
Alice

Hello Alice!

Decrypt

Alice's private key

Alice proves this message is from her



Authentication & Authorization in ASP.NET

- Parts of the app may be public, and parts may be protected to only authorized users
- Controllers or Actions can be protected with an Authorization attribute
- Some actions may require the user to be in a particular role
- There are different types of authentication
  - We are going to use Session-based today
  - In Module 4, we will use token-based

Let’s Code