MANCHESTER 1824
The University of Manchester

Security in Web Applications

Mustafa A. Mustafa
Research Fellow in Computer Science

MANCHESTER 1824
The University of Manchester

Before we start …

What do we want to protect?

Our assets → **Web page, customer data, reputation, …**

Who do we want to protect against?

Potential threats/attackers → **Malicious users:** script kiddies, own customers, competitors, (hostile) foreign state, …

How can our assets be attacked?

Potential vulnerabilities → **Many!!!**

MANCHESTER 1824
The University of Manchester

OWASP top 10 security risks

Credits: neoscopeit.com

MANCHESTER 1824
The University of Manchester

OWASP top 10 security risks

Credits: neoscopeit.com

MANCHESTER 1824
The University of Manchester

Authentication

What is authentication?

It is the process of **confirming a user's identity**.

It protects against **spoofing identity attacks**.

How is it done?

Credentials provided **by users** are **compared to** those **held in a secure user database** (Local operating system or Authentication server).

What types of credentials are there?

Credentials based on **who we are**, **what we know** and **what we have**.

MANCHESTER 1824
The University of Manchester

Examples of credentials

Who you are

- Biometrics
- Fingerprints
- Face Recognition

What you know

- username / password
- PIN (Personal Identification Number) code

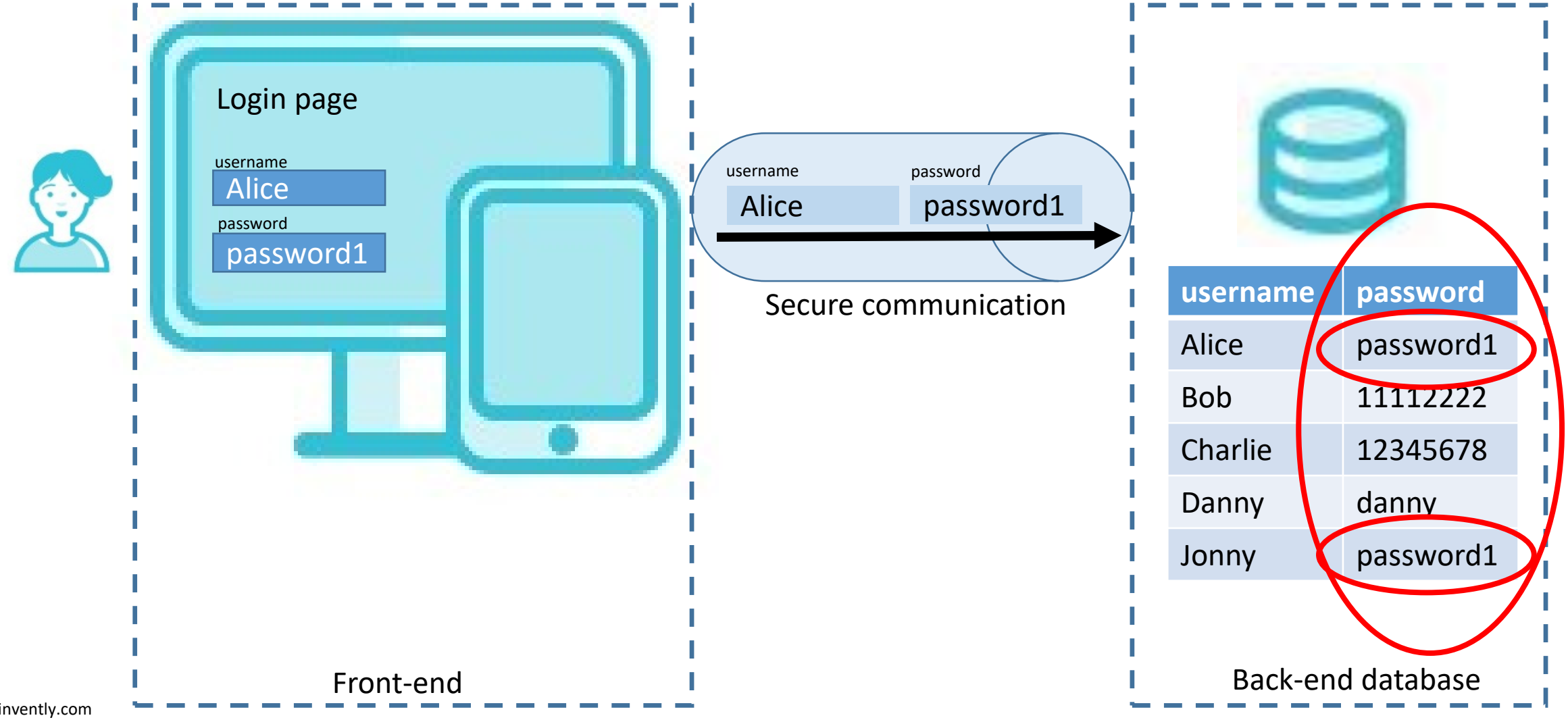
What you have

- RFID card
- USB token
- One-time password token

Used in combination: two-factor authentication (2FA)

MANCHESTER 1824
The University of Manchester

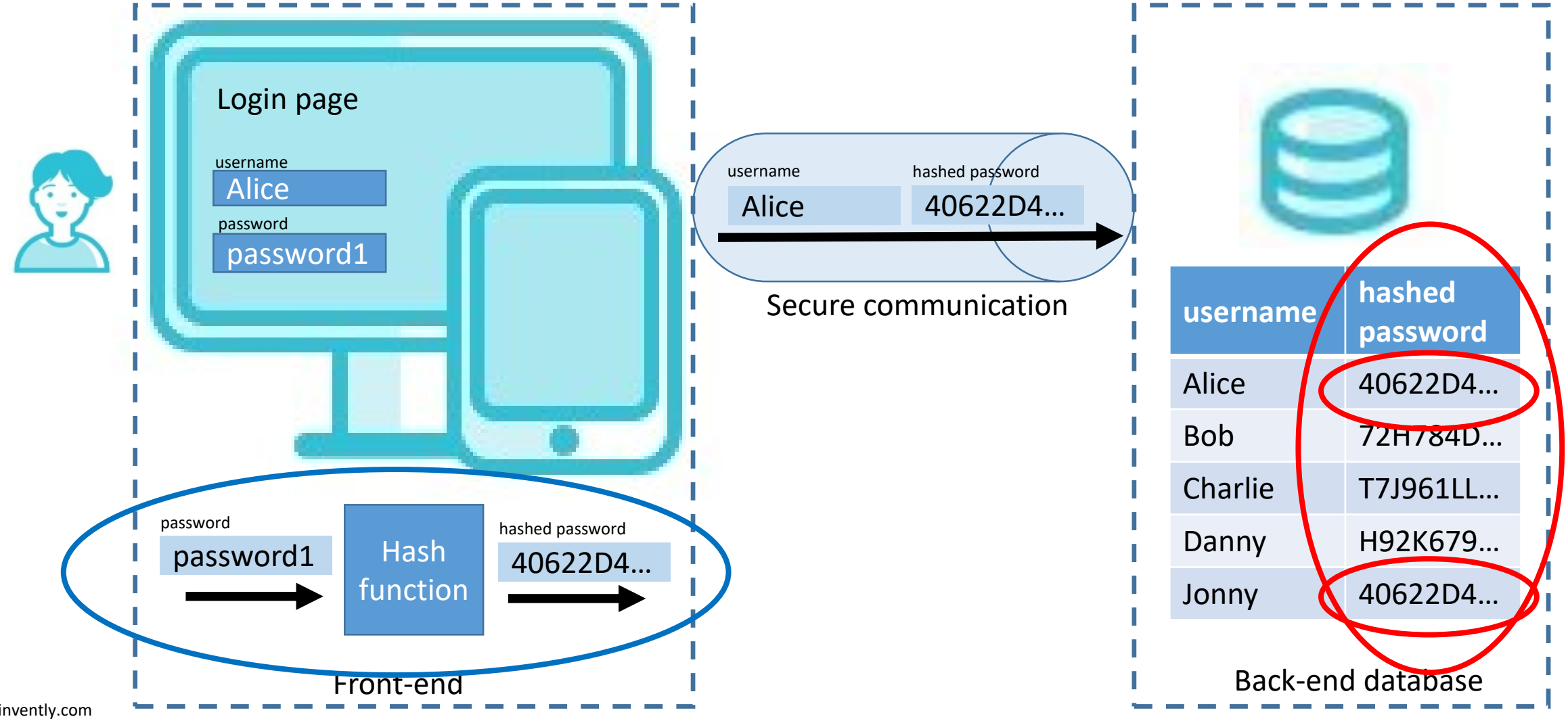
Insecure authentication



Credits: Reinvently.com

MANCHESTER 1824
The University of Manchester

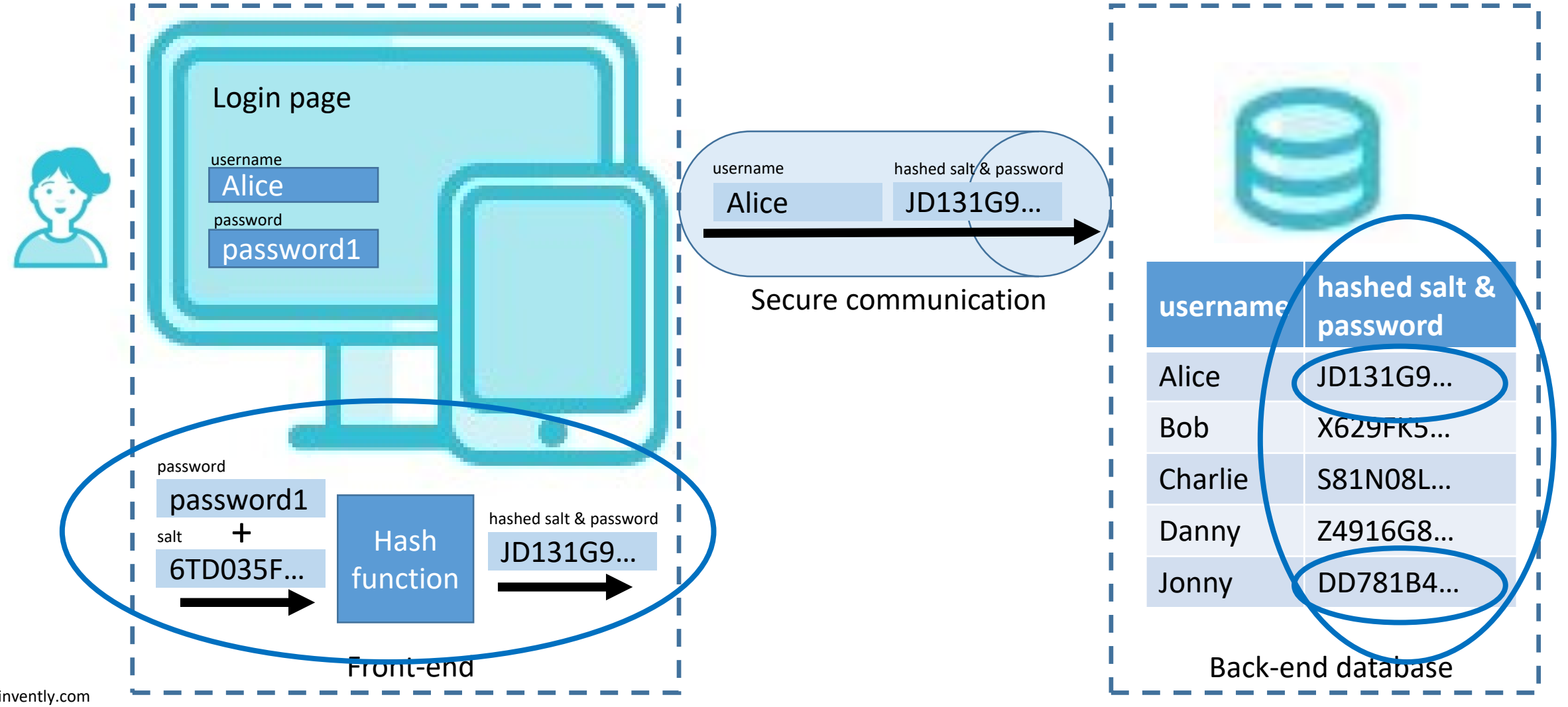
Still insecure authentication



Credits: Reinvently.com

MANCHESTER 1824
The University of Manchester

Secure authentication



Credits: Reinvently.com

MANCHESTER 1824
The University of Manchester

Secure authentication

Example hash functions
MD5 – insecure
SHA1 – not recommended
SHA2 – secure
SHA3 – secure

Example password-hashing functions
bcrypt – secure
scrypt – secure
Lyra2 – secure
Argon2 – secure

MANCHESTER 1824
The University of Manchester

Authorisation

What is authorisation?

It is the process of **specifying access rights/privileges to resources**.

It protects against **privilege escalation attacks**.

What resources are there?

System resources (files, services, computer programs)

Customer data

Application features (delete, add)

How is authorisation done?

Often **role-based** authorisation is used

MANCHESTER 1824
The University of Manchester

Examples of roles and privileges

Admin – can access everything

HR Staff – can access employee records; other relevant documents

Staff – can see own employee record, but no one else's

Guest – very limited access; authentication generally not used/required

MANCHESTER 1824
The University of Manchester

Injection attacks

How they work?

An attacker **injects untrusted input** which gets processed as part of a command or query.

What types are there?

Cross-site request forgery (CSRF); **Cross-site scripting** (XSS); **SQL injection**; …

What is the main reason?

Insufficient user input validation

How to avoid injection attacks?

Never trust user input.

The more you **restrict**, **control**, and **monitor** any form of **user input**, the more you can avoid your application being hacked.

MANCHESTER 1824
The University of Manchester

Implementing security in Spring

Spring supports all major Web security practices

- Authentication
  - Form-based login (username / password) – via session cookies
  - HTTP Basic
- Authorisation
  - Roles
- Cross-site Request Forgery Protection

Provided by the Spring Security project

- See the Spring Security documentation for details

MANCHESTER 1824
The University of Manchester

Security in EventLite

- See the Security class in the config package
- @EnableWebSecurity turns on the security system
- Fine-tune security configuration by overriding methods in WebSecurityConfigurerAdapter
- Current configuration:
  - Log in required for all addresses, with exceptions
    - GETs, h2-console
- Same users as week 1 code
  - Rob, Caroline, Markel, Mustafa
  - Stored in memory
  - All users have the “ADMINISTRATOR” role
- HTTP Basic authentication for API

MANCHESTER 1824
The University of Manchester

Example code from week 1

```
// Create an admin role
public static final String ADMIN_ROLE = "ADMINISTRATOR";
```

```
// List the mappings/methods for which no authorisation is required.
// This includes the paths where static resources, such as bootstrap, are
// located.
private static final RequestMatcher[] NO_AUTH = {
new AntPathRequestMatcher("/webjars/**", "GET"),
new AntPathRequestMatcher("/", "GET"),
new AntPathRequestMatcher("/api/**", "GET"),
new AntPathRequestMatcher("/greeting", "GET"),
new AntPathRequestMatcher("/greeting/{id:[\\d]+}", "GET") };
```

MANCHESTER 1824
The University of Manchester

Example code from week 1

```
@Override
protected void configure(HttpSecurity http) throws Exception {

        // By default, all requests are authenticated except our specific list.
        http.authorizeRequests().requestMatchers(NO_AUTH).permitAll().anyRequest().
hasRole(ADMIN_ROLE);

        // Use form login/logout for the Web.
        http.formLogin().loginPage("/sign-in").permitAll();
        http.logout().logoutUrl("/sign-out").logoutSuccessUrl("/").permitAll();

        // Use HTTP basic for the API.
        http.requestMatcher(new AntPathRequestMatcher("/api/**")).httpBasic();

        // Only use CSRF for Web requests.
        http.antMatcher("/**").csrf().ignoringAntMatchers("/api/**");
        }
```

MANCHESTER 1824
The University of Manchester

Example code from week 1

```
@Override
@Bean
public UserDetailsService userDetailsService() {
	PasswordEncoder encoder =
PasswordEncoderFactories.createDelegatingPasswordEncoder();

	UserDetails rob =
User.withUsername("Rob").password(encoder.encode("Haines")).roles(ADMIN_ROLE).build();
	UserDetails markel =
User.withUsername("Markel").password(encoder.encode("Vigo")).roles(ADMIN_ROLE).build();

	return new InMemoryUserDetailsManager(rob, markel);
	}
```

MANCHESTER 1824
The University of Manchester

Thank you!

Mustafa A. Mustafa
mustafa.mustafa@manchester.ac.uk