How Your Website Handles Secure Data

by Dan Ficker

May 21 - MidCamp 2025

Dan Ficker

Staff CS Engineer, Pantheon

- Lives in St. Paul, MN
- Does mostly backend development, but likes site building with Drupal too
- Nearly 20 years using Drupal
- Over 25 years since I built my first website
- Enjoys technology, music and movies

How Your Website Handles Secure Data

Overview

What is Security?

Do I Have To Worry About Security?

Transmitting Secure Data

Storing Secure Data

Protecting Your Site From Attacks

What Is Security?

The Basics

- Some pages on your site are public. But not everything is public.
- Security is keeping the private data private.
- Only give out the private data to the correct users.
- Malicious or unwanted visitors cannot access secure data.

What Is Security?

Security Is Easy, Right?

- Yes: Security is built into modern web servers, browsers and Drupal.
- Yes: Security best practices & robust Open Source libraries are available.
- No: New issues could be found with any software at any time.
- No: Custom code should be written with security in mind.

Yes!

You Should Be Concerned About Security

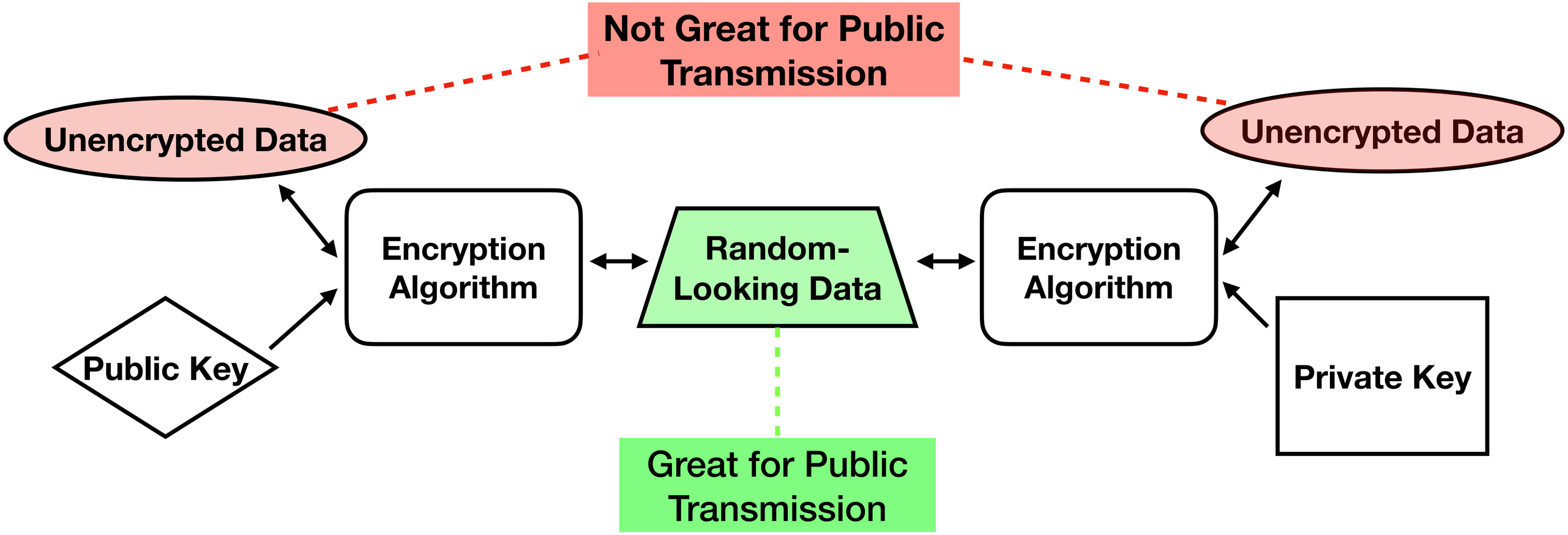
Security Concept

Public Key Encryption

- Two keys are generated and work together.
- The Private Key must be secret while the Public Key can be given freely.
- Public Key can decrypt messages encrypted with the Private Key.
- Public Key can encrypt messages that can only be decrypted with the Private Key.

Public Key Encryption

A Diagram



Public Key Encryption

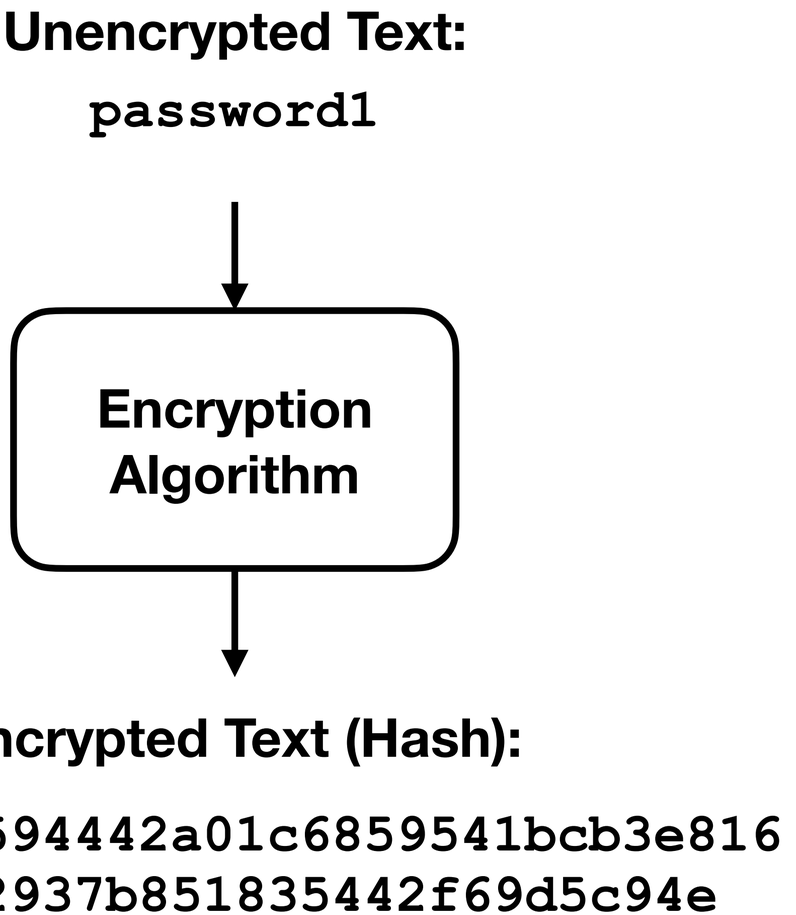
Common Uses Cases

- HTTPS Web Pages (SSL/TLS):
  - The Web Server has a Private Key
  - When your device connects, the Server gives you the Public Key
  - This is done at the Web Server Level (Apache/Nginx, before Drupal)
- SSH Keys
  - Your local SSH client has a Private Key, and you give GitHub and other services the Public Key.
  - Bonus: Can be used as authentication method too.

Security Concept

One-Way Encryption

- Also known as “hashing”, a process of turning some text into some other text that is indecipherable from random data.
- The process is irreversible—there’s no way to get back to the original data if you only know the end result.
- No matter what is put in, a hash of the same length is returned. There is a possibility of collisions (the same hash coming from different inputs).
- Modifying one character of the original text will change the hash drastically.



One-Way Encryption

Passwords In Drupal

- Drupal does not store account passwords in plain text.
- It hashes the password before it is stored in the database.
- Modern Drupal defaults to using `sha512` hash.
- Supports older `md5` hash for backwards compatibility.

```php
abstract class PhpassHashedPasswordBase implements PasswordInterface {

  /**
   * {@inheritdoc}
   */
  public function hash(#[\SensitiveParameter] $password) {
    if (isset($this->corePassword)) {
      return $this->corePassword->hash($password);
    }

    return $this->crypt('sha512', $password, $this->generateSalt());
  }

  /**
   * {@inheritdoc}
   */
  public function check(#[\SensitiveParameter] $password, #[\SensitiveParame
    // Newly created accounts may have empty passwords.
    if ($hash === NULL || $hash === '') {
      return FALSE;
    }
    if (str_starts_with($hash, 'U$')) {
      // This may be an updated password from user_update_7000(). Such hashe
      // have 'U' added as the first character and need an extra md5() (see
      // Drupal 7 documentation).
      $stored_hash = substr($hash, 1);
      $password = md5($password);
    }
    else {
      $stored_hash = $hash;
    }
```

One-Way Encryption

Passwords In Drupal (cont.)

- Drupal “salts” the password: adds some unique data to each password hash to make it harder to crack passwords.
- Drupal literally does not know your password. It just does the same hash process when you login with a password and verifies if the hash matches or not. If it matches, you get a logged in session.

```php
abstract class PhpassHashedPasswordBase implements PasswordInterface {
  protected function crypt($algo, #[\SensitiveParameter] $password, $setting) {
    // Prevent DoS attacks by refusing to hash large passwords.
    if (strlen($password) > PasswordInterface::PASSWORD_MAX_LENGTH) {
      return FALSE;
    }

    // The first 12 characters of an existing hash are its setting string.
    $setting = substr($setting, 0, 12);

    if ($setting[0] != '$' || $setting[2] != '$') {
      return FALSE;
    }
    $count_log2 = $this->getCountLog2($setting);
    // Stored hashes may have been encrypted with any iteration count. However
    // we do not allow applying the algorithm for unreasonable low and high
    // values respectively.
    if ($count_log2 != $this->enforceLog2Boundaries($count_log2)) {
      return FALSE;
    }
    $salt = substr($setting, 4, 8);
    // Hashes must have an 8 character salt.
    if (strlen($salt) != 8) {
      return FALSE;
    }

    // Convert the base 2 logarithm into an integer.
    $count = 1 << $count_log2;

    $hash = hash($algo, $salt . $password, TRUE);
    do {
      $hash = hash($algo, $hash . $password, TRUE);
    } while (--$count);
```

Storing Sensitive Data In Drupal

- Try not to store sensitive data if possible. Storing sensitive data makes your site a bigger target.
- Security experts have created methods for encrypting and decrypting data as securely as possible. Use the standards. Don’t try to build your own secure data encryption system. You may miss something important that makes it easy for hackers to get your data.
- Hackers think a lot different than a developer. It’s really hard to develop secure software because we cannot think of every way to exploit it.
- Encrypt Module (https://www.drupal.org/project/encrypt) provides APIs for encrypting of sensitive data using trusted standards.

Security In Drupal

General Guidance

- https://www.drupal.org/security - Latest Info and Resources
- When browsing modules, look for the shield icon. These are stable and more trusted releases.
- Security Kit Module: https://www.drupal.org/project/seckit Helps add headers that can be helpful for improving security.
- Keep your site's core, modules and themes updated!

Thanks!
Any Questions?

Thanks To A Podcast

Security Now

- This podcast covers tech and security news very well.
- It has inspired me to talk about security and taught me many of these topics.
- Has nearly 1,000 episodes spanning the last 19+ years.
- Early episodes cover the basics of security, the Internet, etc.