PHP: Bezpieczne przesyłanie plików w PHP

aplikacja dostępna pod linkiem: https://wiki.ostrowski.net.pl/copy-file/

Strona powitalna

- Po wejściu na index.php użytkownik widzi dwa przyciski: „I’m Sender” i „I’m Receiver”.
- Wybierając jedną z opcji, przechodzi do odpowiedniego trybu — nadawcy lub odbiorcy.

Generowanie kodu („Sender”)

- Gdy klikniesz „I’m Sender”, aplikacja losuje przyjazny kod w postaci dwóch członów: przymiotnik-zwierzę (np. brave-fox, calm-owl).
- Ten kod nie jest samą nazwą katalogu, a jedynie ziarnem (seedem) do dalszych kroków.

Przypisanie katalogu

- Na podstawie unikalnego kodu tworzone jest losowe ID katalogu (np. a3f1b5c7d9e2f0a1...), zapisane w pliku mappings/<kod>.
- Jednocześnie powstaje fizyczny katalog uploads/<ID>/, w którym będą przechowywane zaszyfrowane pliki.

Derywacja klucza szyfrującego

- Klucz AES-256 do szyfrowania/dekryptowania wyprowadzany jest w locie z kodu użytkownika: SHA-256(kod) → bity → klucz (32 bajty).
- Dzięki temu klucz nigdy nie jest zapisywany na dysku.

Wysyłanie pliku (upload)

Nadawca wybiera plik i wysyła go formularzem.

Aplikacja:

- Odczytuje oryginalny plik z tymczasowego miejsca.
- Generuje losowy wektor inicjalizujący (IV) 16 bajtów.
- Szyfruje zawartość pliku algorytmem AES-256-CBC (używając klucza i IV).
- Zapisuje do uploads/<ID>/<nazwa_pliku>.enc zawartość: [IV][szyfrogram].

Wyświetlenie kodu

Po udanym uploadzie nadawca widzi komunikat ze swoim kodem (np. brave-fox) i może go przekazać odbiorcy.

Usuwanie danych po zamknięciu (cleanup)

W widoku nadawcy skrypt JavaScript nasłuchuje zdarzenia unload (zamykania/odświeżania strony).

Jeśli użytkownik nie klika właśnie „Wyślij” (czyli naprawdę opuszcza kartę), wysyłany jest lekki sygnał (navigator.sendBeacon) z parametrem mode=cleanup&code=<kod>.

Na serwerze obsługa tej akcji usuwa katalog uploads/<ID>/ i plik mapujący mappings/<kod>.

Odbieranie pliku (Receiver)

W trybie odbiorcy pojawia się formularz, w który trzeba wpisać otrzymany od nadawcy kod (np. brave-fox).

Po zatwierdzeniu:

- Aplikacja odczytuje z pliku mappings/brave-fox odpowiadające mu ID katalogu.
- Przegląda zawartość uploads/<ID>/ i listuje wszystkie pliki .enc.
- Ściąganie pliku (download)

Odbiorca klika „Download” przy wybranym zaszyfrowanym pliku.

Serwer:

- Odczytuje uploads/<ID>/<plik>.enc.
- Oddziela IV (pierwsze 16 bajtów) od szyfrogramu.
- Dekryptuje zawartość tym samym kluczem wyprowadzonym z kodu.
- Wysyła odeszyfrowany plik w odpowiedzi jako załącznik.

Powtórne użycie i bezpieczeństwo

- Odbiorca nie przechowuje kodu w sesji — za każdym razem musi go wpisać od nowa.
- Katalogi na serwerze mają losowe nazwy, niepowiązane jawnie z kodem, co zapobiega odgadnięciu klucza.
- Klucz szyfrujący nigdy nie trafia na dysk — jest zawsze generowany w pamięci.

Dzięki tej procedurze:

- Bezpieczeństwo: klucz nie zapisuje się w plikach, kod nie ujawnia nazwy katalogu.
- Łatwość użycia: prosty, czytelny kod typu adjective-animal.
- Automatyczne czyszczenie: nadawca nie pozostawia śladów po zamknięciu sesji.

Kod Rozwiązania

index.php

```php
<?php
// index.php
// ──────────────────────────────────────────────────
// CONFIGURATION
define('UPLOAD_DIR',  __DIR__ . '/uploads/');
define('MAP_DIR',     __DIR__ . '/mappings/');
define('IV_LEN',      16);                // AES block size
define('DIR_ID_LEN',  16);                // bytes for random dir ID

$adjectives = ['lista','przymiotników'];
// longer the better but still bruteforcable if you have 100 adjectives
and animals
//you get 10k possiblities which is not that much, but anyway it is
better then
//retyping 32 bit random codes or something ;-)
$animals    = ['lista', 'zwierząt'];

// Ensure dirs exist
foreach ([UPLOAD_DIR, MAP_DIR] as $d) {
    if (!is_dir($d)) mkdir($d, 0700, true);
}

// ──────────────────────────────────────────────────
// HELPERS

function make_code(array $a, array $b): string {
    return $a[array_rand($a)] . '-' . $b[array_rand($b)];
}

function derive_key(string $code): string {
    return hex2bin(hash('sha256', $code));  // 32 bytes
}

function encrypt_file(string $src, string $dst, string $code): bool {
    $key = derive_key($code);
    $iv  = random_bytes(IV_LEN);
    $pt  = file_get_contents($src);
    $ct  = openssl_encrypt($pt, 'AES-256-CBC', $key, OPENSSL_RAW_DATA,
$iv);
    return $ct === false
        ? false
        : file_put_contents($dst, $iv . $ct) !== false;
}

function decrypt_file(string $path, string $code) {
    $key  = derive_key($code);
```

```
    $data = file_get_contents($path);
    $iv   = substr($data, 0, IV_LEN);
    $ct   = substr($data, IV_LEN);
    return openssl_decrypt($ct, 'AES-256-CBC', $key, OPENSSL_RAW_DATA,
$iv);
}

function rrmdir(string $dir) {
    if (!is_dir($dir)) return;
    foreach (scandir($dir) as $f) {
        if ($f==='.'||$f==='..') continue;
        $p = "$dir/$f";
        is_dir($p) ? rrmdir($p) : unlink($p);
    }
    rmdir($dir);
}

function valid_code(string $c): bool {
    return preg_match('/^[a-z]+-[a-z]+$/', $c) === 1;
}

// Lookup directory ID by code
function get_dir_by_code(string $code): ?string {
    $map = MAP_DIR . $code;
    if (!file_exists($map)) return null;
    return trim(file_get_contents($map));
}

// Create mapping code→random dir ID
function create_mapping(string $code): string {
    // generate a random hex ID
    $dirId = bin2hex(random_bytes(DIR_ID_LEN));
    // create upload dir
    mkdir(UPLOAD_DIR . $dirId, 0700, true);
    // write mapping file (only dirId)
    file_put_contents(MAP_DIR . $code, $dirId);
    return $dirId;
}

// Remove mapping + uploads
function cleanup(string $code) {
    $dirId = get_dir_by_code($code);
    if ($dirId) {
        rrmdir(UPLOAD_DIR . $dirId);
        unlink(MAP_DIR . $code);
    }
}

// ——————————————————————————————————————————
// ROUTING
$mode = $_REQUEST['mode'] ?? null;
```

```php
$code = $_REQUEST['code'] ?? null;

// 1) CLEANUP on unload
if ($mode==='cleanup' && valid_code($code)) {
    cleanup($code);
    http_response_code(204);
    exit;
}

// 2) DOWNLOAD (receiver)
if ($mode==='download' && valid_code($code) && isset($_GET['file'])) {
    $dirId = get_dir_by_code($code);
    $file  = basename($_GET['file']);
    $path  = UPLOAD_DIR . "$dirId/$file";
    if (!$dirId || !file_exists($path)) {
        http_response_code(404);
        exit('Not found.');
    }
    $data = decrypt_file($path, $code);
    if ($data === false) {
        http_response_code(500);
        exit('Decrypt error.');
    }
    header('Content-Disposition: attachment;
filename="'.pathinfo($file,PATHINFO_FILENAME).'"');
    header('Content-Type: application/octet-stream');
    echo $data;
    exit;
}

// 3) SENDER UPLOAD
if ($_SERVER['REQUEST_METHOD']==='POST'
 && $_POST['mode']==='sender'
 && !empty($_POST['code'])
) {
    $code = $_POST['code'];
    if (!valid_code($code)) {
        $error = "Invalid code.";
    } else {
        // get or create a dir ID for this code
        $dirId = get_dir_by_code($code) ?? create_mapping($code);

        if (empty($_FILES['file']) ||
$_FILES['file']['error']!==UPLOAD_ERR_OK) {
            $error = "Upload error.";
        } else {
            $orig = $_FILES['file']['tmp_name'];
            $name = basename($_FILES['file']['name']) . '.enc';
            $dst  = UPLOAD_DIR . "$dirId/$name";
            if (encrypt_file($orig, $dst, $code)) {
                $success = "File encrypted & uploaded as
```

```
<b>{$name}</b>.<br>Share this code:<br><b>{$code}</b>";
            } else {
                $error = "Encryption failed.";
            }
        }
    }
}

// 4) RECEIVER ENTER CODE
if ($_SERVER['REQUEST_METHOD']==='POST'
 && $_POST['mode']==='receiver'
 && !empty($_POST['code'])
) {
    $code = $_POST['code'];
    if (!valid_code($code)) {
        $error = "Invalid code.";
    }
    // else $code is used for this request only
}

// ____________________________________________________
// PAGE
?><!DOCTYPE html>
<html lang="en"><head><meta charset="UTF-8">
  <title>Secure File Exchange</title>
  <style>
    body { font-family:sans-serif; background:#f4f4f9; margin:0;
padding:20px; }
    .box { background:#fff; padding:20px; border-radius:8px;
           max-width:400px; margin:40px auto; box-shadow:0 2px 6px
rgba(0,0,0,0.1); }
    input,button,a { display:block; width:100%; margin:8px 0; }
    .error{color:#c00} .success{color:#080}
  </style>
</head><body>
  <div class="box">
    <h1>Secure File Exchange</h1>

  <?php if (!$mode): // Landing ?>
    <a href="?mode=sender"><button>I'm Sender</button></a>
    <a href="?mode=receiver"><button>I'm Receiver</button></a>

  <?php elseif ($mode==='sender'): // Sender UI ?>
    <h2>Sender</h2>
    <?php if (!isset($code)): $code = make_code($adjectives,$animals);
endif; ?>
    <?php if (!empty($error)): ?><p class="error"><?= $error
?></p><?php endif; ?>
    <?php if (!empty($success)): ?><p class="success"><?= $success
?></p><?php endif; ?>
```

```
    <p>Your code:<br><b><?= htmlspecialchars($code) ?></b></p>
    <form id="uploadForm" method="post" enctype="multipart/form-data">
      <input type="hidden" name="mode"  value="sender">
      <input type="hidden" name="code"  value="<?=
htmlspecialchars($code) ?>">
      <input type="file" name="file" required>
      <button type="submit">Upload & Encrypt</button>
    </form>
    <a href="index.php">← Back</a>

    <script>
      let submitting = false;
      document.getElementById('uploadForm')
        .addEventListener('submit', ()=>{ submitting = true; });
      window.addEventListener('unload', ()=>{
        if (!submitting) {
          navigator.sendBeacon('?mode=cleanup&code=<?= $code ?>');
        }
      });
    </script>

  <?php elseif ($mode==='receiver'): // Receiver UI ?>
    <h2>Receiver</h2>
    <?php if (empty($code) || !empty($error)): ?>
      <form method="post">
        <input type="hidden" name="mode"  value="receiver">
        <input type="text"   name="code"  placeholder="Enter code"
required>
        <button type="submit">Submit</button>
      </form>
      <?php if (!empty($error)): ?><p class="error"><?= $error
?></p><?php endif; ?>

    <?php else:
      $dirId = get_dir_by_code($code);
      $files = $dirId
             ? array_diff(scandir(UPLOAD_DIR . $dirId), ['.','..'])
             : [];
      if (empty($files)): ?>
        <p>Waiting for sender…</p>
      <?php else: ?>
        <ul>
          <?php foreach ($files as $f): ?>
            <li>
              <?= htmlspecialchars(pathinfo($f,PATHINFO_FILENAME)) ?>
              — <a href="?mode=download&code=<?=
htmlspecialchars($code) ?>&file=<?= urlencode($f) ?>">Download</a>
            </li>
          <?php endforeach; ?>
        </ul>
      <?php endif; ?>
```

```
      <a href="index.php?mode=receiver">← New code</a>
    <?php endif; ?>

  <?php else: // Invalid mode ?>
    <p class="error">Invalid mode.</p>
    <a href="index.php">Back</a>
  <?php endif; ?>

  </div>
</body></html>
```