Supplementary Material for: Beyond Mod-97: Phase-Coupled $D_5 \times D_5$ Folds for Two-Digit Decimal Checksums with Exact Bounds

Csaba Balogh

Online Resource 1: Worked Example

Let the payload be $\mathbf{p} = (1, 2, 3)$ and the recommended phase $k = 4$. Both layers of the contribution map are obtained from the cycle-conjugate permutation: layer 1 uses $\pi_i(d) = \rho[(\rho^{-1}(d) + i) \bmod 10]$ and layer 2 uses $\pi_{i+k}(d) = \rho[(\rho^{-1}(d) + i + 4) \bmod 10]$, with $\rho = [0, 1, 4, 3, 8, 9, 6, 7, 2, 5]$ and $\rho^{-1} = [0, 1, 8, 3, 2, 9, 6, 7, 4, 5]$. We trace the payload syndrome accumulation step by step.

i	p_j	$\pi_i(p_j)$	$\pi_{i+4}(p_j)$	S_1 after $\mathrm{Mul}(S_1, \varphi(\pi_i))$	S_2 after $\mathrm{Mul}(S_2, \varphi(\pi_{i+4}))$	Note
2	1	3	7	$(3, 0)$	$(2, 1)$	$\varphi(3) = (3, 0)$, $\varphi(7) = (2, 1)$
3	2	1	9	$(4, 0)$	$(3, 0)$	S_2: $(2, 1) \otimes (4, 1) = (2 - 4 \bmod 5,\ 0) = (3, 0)$
4	3	7	1	$(1, 1)$	$(4, 0)$	S_1: $(4, 0) \otimes (2, 1) = (4 + 2 \bmod 5,\ 1) = (1, 1)$

Final syndrome: $P^{(4)}(\mathbf{p}) = \big((1, 1),\ (4, 0)\big)$. Inverse: $(1, 1)^{-1} = (1, 1)$ (a reflection is its own inverse) and $(4, 0)^{-1} = (1, 0)$. Therefore $(c_1, c_2) = \big(\varphi^{-1}(1, 1),\ \varphi^{-1}(1, 0)\big) = (6, 1)$.

The receiver, given the codeword $(6, 1, 1, 2, 3)$, embeds the check digits through $\eta(6, 1) = \big((1, 1),\ (1, 0)\big)$ and folds the full sequence; the result is $\big((0, 0),\ (0, 0)\big)$, confirming validity.

Now suppose positions 3 and 4 are transposed, changing the codeword from $(6, 1, 1, 2, 3)$ to $(6, 1, 1, 3, 2)$. Recomputing the fold yields a non-identity element of G and the validation step rejects the input, consistent with Theorem 6 of the main paper.

Online Resource 2: Reference Pseudocode

The two listings below translate the algebraic definitions of the main paper (Sections II and VI) into executable form. The group operations reduce to modular-5 arithmetic plus a single XOR; the position-dependent permutation π_i uses two constant vectors ρ, ρ^{-1} and a single modular-10 addition (no lookup tables, no conditional branching).

Listing 1: Primitive operations on $S = \mathbb{Z}_5 \rtimes \mathbb{Z}_2$, the cycle-conjugate permutation π_i, and the two-layer contribution map $\Psi_i^{(k)}$.

```
constant rho     := [0, 1, 4, 3, 8, 9, 6, 7, 2, 5]
constant rho_inv := [0, 1, 8, 3, 2, 9, 6, 7, 4, 5]

function Mul((a,b), (c,d)):              -- D_5 product
    sign := +1 if b = 0, else -1
    return ( (a + sign * c) mod 5,  b XOR d )

function Inv((a,b)):                     -- D_5 inverse
    if b = 0: return ( (-a) mod 5, 0 )
    else:     return ( a, 1 )            -- reflection: self-inverse

function Phi(d):                         -- decimal -> D_5
    if d < 5: return (d, 0)
    else:     return (d - 5, 1)

function PhiInv((a,b)):                  -- D_5 -> decimal
    if b = 0: return a
    else:     return a + 5

function Pi(d, i):                       -- cycle-conjugate permutation
    return rho[ (rho_inv[d] + i) mod 10 ]

function Psi(d, i, k):                   -- two-layer contribution in G
    layer1 := Phi(Pi(d, i))
```

```
    layer2 := Phi(Pi(d, i + k))
    return ( layer1, layer2 )

function MulG((g1,g2), (h1,h2)):          -- componentwise product on G = D_5 x D_5
    return ( Mul(g1, h1), Mul(g2, h2) )

function InvG((g1,g2)):                   -- componentwise inverse on G
    return ( Inv(g1), Inv(g2) )
```

Listing 2: Encoding of two decimal check digits and validation of a received codeword. Positions 0 and 1 are reserved for the check digits; the payload occupies positions 2 through $m + 1$.

```
function ComputeCheckDigits(p[0..m-1], k):
    -- Recommended k = 4 (Section VIII of the main paper).
    P := ( (0,0), (0,0) )                 -- identity in G
    for j from 0 to m-1:
        P := MulG(P, Psi(p[j], j + 2, k))
    P_inv := InvG(P)                      -- componentwise inversion
    c1 := PhiInv( P_inv[1] )              -- layer-1 inverse  -> first check digit
    c2 := PhiInv( P_inv[2] )              -- layer-2 inverse  -> second check digit
    return ( c1, c2 )

function Validate(c1, c2, p[0..m-1], k):
    -- Returns TRUE iff the received codeword (c1, c2, p[0], ..., p[m-1])
    -- satisfies the fold-closure condition.
    eta := ( Phi(c1), Phi(c2) )           -- direct embedding of check digits
    P := eta
    for j from 0 to m-1:
        P := MulG(P, Psi(p[j], j + 2, k))
    return P = ( (0,0), (0,0) )
```

The encoder runs in $\mathcal{O}(m)$ time with $\mathcal{O}(1)$ extra memory beyond the two constant vectors ρ, ρ^{-1}. The decoder has the same complexity. No table of 190 entries (as in the classical Verhoeff formulation) is needed, and no non-decimal symbol can ever appear in the output.

Online Resource 3: Additional Monte Carlo Data

Table S1 reports detection rates from a smaller pilot run ($n = 5{,}000$ trials, seed 12345) for three representative non-degenerate phases. Results are consistent with the $n = 100{,}000$ run reported in Table II of the main text. Confidence intervals are Wilson 95% intervals.

TABLE I
*

Table S1: Detection rates from Monte Carlo runs with $n = 5{,}000$ trials per configuration (seed 12345, codeword length 10). Wilson 95% intervals in the last two columns. The twin-error class has fewer than 5,000 *tested* cases because not every random codeword contains an adjacent twin; only those that do are counted.

Error class	k	Trials	Detected	Rate	CI low	CI high
Twin error ($aa \to bb$)	1	2981	2981	1.0000	0.9987	1.0000
Jump transposition	1	5000	4962	0.9924	0.9896	0.9945
Two-digit substitution	1	5000	4922	0.9844	0.9806	0.9875
Random multi-position noise	1	5000	4978	0.9956	0.9933	0.9971
Twin error ($aa \to bb$)	4	3076	3076	1.0000	0.9988	1.0000
Jump transposition	4	5000	4980	0.9960	0.9938	0.9974
Two-digit substitution	4	5000	4923	0.9846	0.9808	0.9877
Random multi-position noise	4	5000	4980	0.9960	0.9938	0.9974
Twin error ($aa \to bb$)	6	3060	3060	1.0000	0.9987	1.0000
Jump transposition	6	5000	4979	0.9958	0.9936	0.9973
Two-digit substitution	6	5000	4920	0.9840	0.9801	0.9871
Random multi-position noise	6	5000	4982	0.9964	0.9943	0.9977

Point estimates are stable across the two sample sizes: the 5,000-trial pilot rates match the 100,000-trial values from Table II of the main paper to within ± 0.4 percentage points on every class, and the Wilson intervals of the pilot all enclose the

corresponding main-text point estimates. The intervals shrink by a factor of approximately $\sqrt{100{,}000/5{,}000} \approx 4.5$ as expected from standard sampling theory.

The pilot data also reproduce the structural twin-error split between $k = 4$ (and $k = 6$, both at 100%) and the avoided phases $k \in \{3, 7\}$ from Table II.