Checksums/error correcting codes

Parity bits

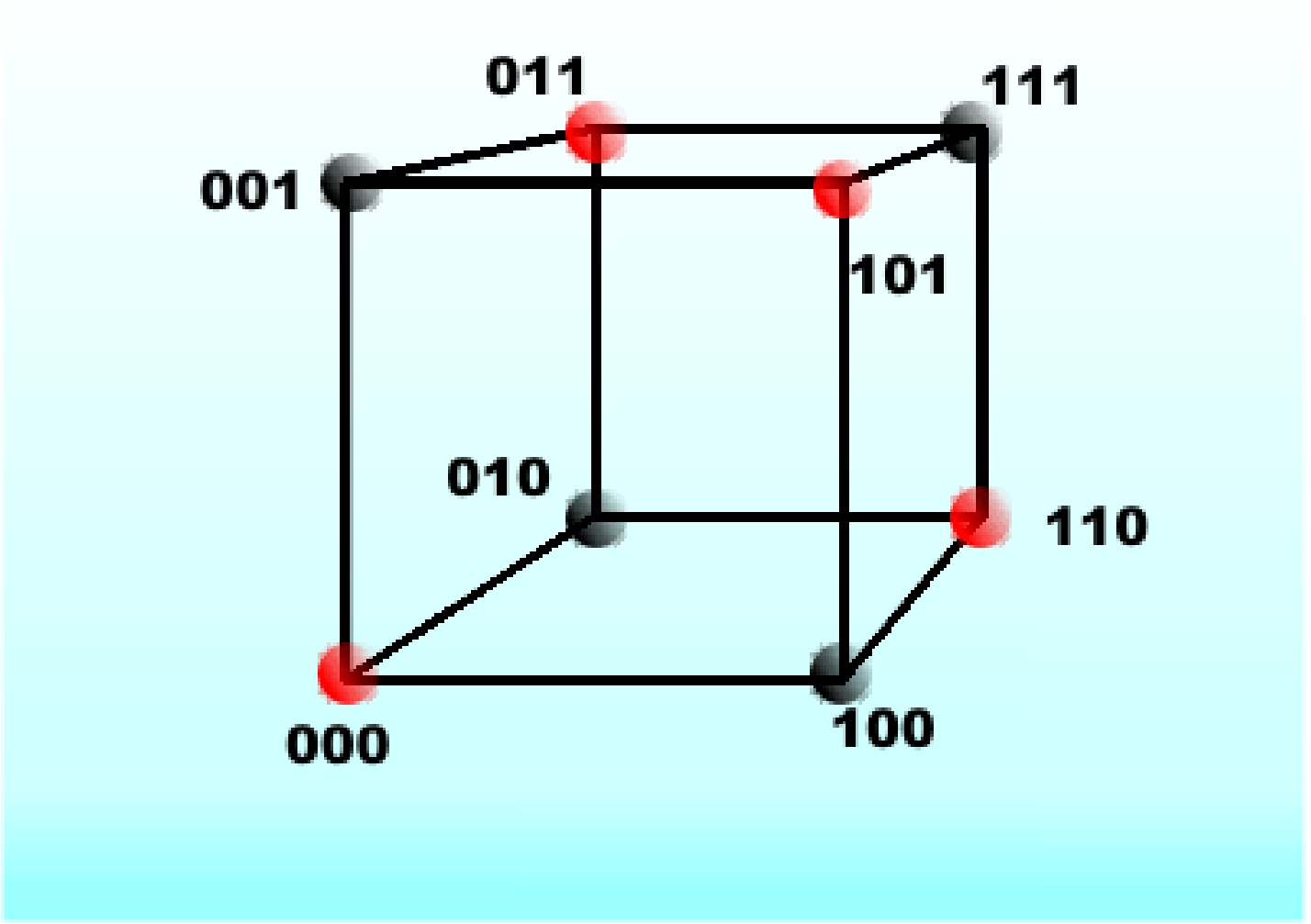
- Add 1 if number of 1s in 'message' is odd, and 0 otherwise

  For 2-bit numbers:

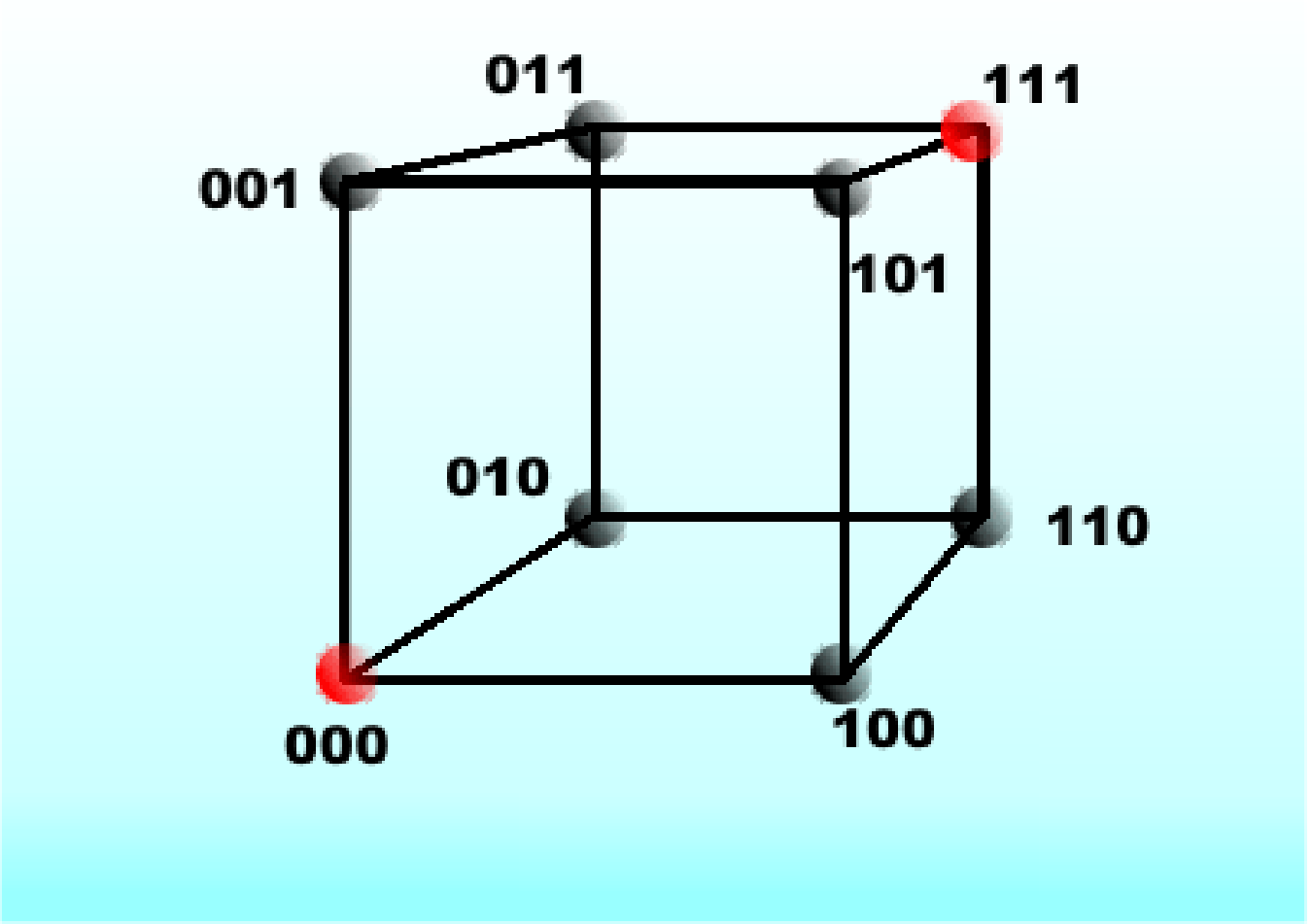
  3 = 11 => 11**0**
  2 = 10 => 10**1**
  ...

- One-bit errors are obvious (even in the parity bit)

Hyper-cube interpretation



Error-correcting codes

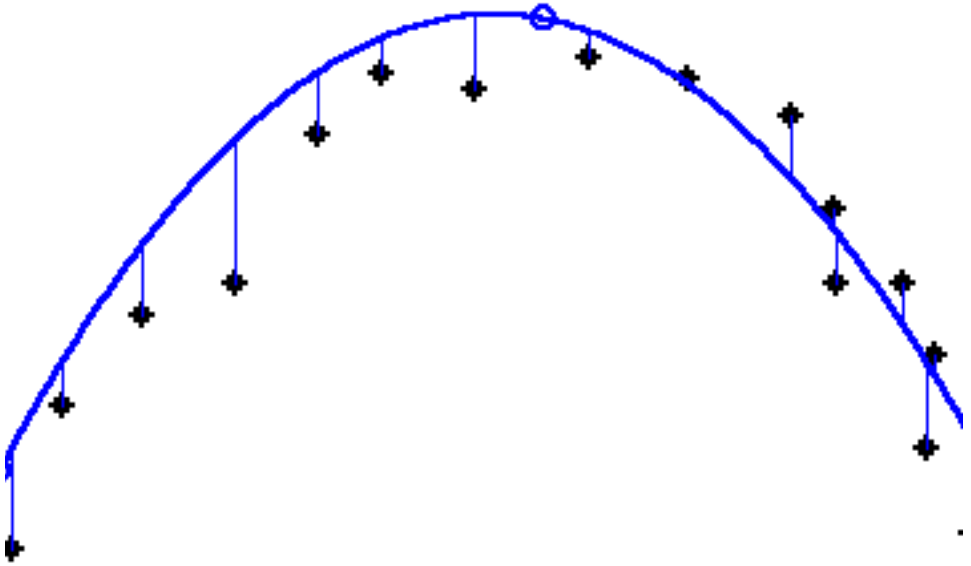


see: http://www.i-programmer.info/babbages-bag/214-error-correcting-codes.html

More generally... complex math

Transform data (add bits) so that it fits a nice mathematical object

Errors become obvious as does a way to correct them.



MD5 Sum

- See: https://tools.ietf.org/html/rfc1321

  Message: $b_0\ b_1 \ldots b_n$
  Padding: $b_0\ b_1 \ldots b_n\ 1\ 0 \ldots 0$ (until length % 512 = 448)
  Append: $b_0\ b_1 \ldots b_n\ 1\ 0 \ldots 0$ [64-bit representation of n]

- 4 registers pre-filled as follows

  word A: 01 23 45 67
  word B: 89 ab cd ef
  word C: fe dc ba 98
  word D: 76 54 32 10

MD5 Sum

- 4 functions on 32-bit words

  F(X,Y,Z) = X and Y or not(X) and Z
  G(X,Y,Z) = X and Z or Y not(Z)
  H(X,Y,Z) = X xor Y xor Z
  I(X,Y, Z) = Y xor (X or not(Z))

- Prepare T[1 ... 64] where T[i] = int (4294967296 * abs(sin(i)))

MD5 Sum

- For each 16-word block (word = 32 bits) in message M

  for i = 0 to N/16 – 1 do

  copy ith block into X

  AA = A

  BB = B

  CC = C

  DD = D

  4-round shuffle of A, B, C, D (next slide)

  A = AA + A

  B = BB + B

  C = CC + C

  D = DD + D

  done

  Output A, B, C, D

MD5 Sum

```
/* Round 1. */
/* Let [abcd k s i] denote the operation
     a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
/* Do the following 16 operations. */
[ABCD  0  7  1]  [DABC  1 12  2]  [CDAB  2 17  3]  [BCDA  3 22  4]
[ABCD  4  7  5]  [DABC  5 12  6]  [CDAB  6 17  7]  [BCDA  7 22  8]
[ABCD  8  7  9]  [DABC  9 12 10]  [CDAB 10 17 11]  [BCDA 11 22 12]
[ABCD 12  7 13]  [DABC 13 12 14]  [CDAB 14 17 15]  [BCDA 15 22 16]

/* Round 2. */
/* Let [abcd k s i] denote the operation
     a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
/* Do the following 16 operations. */
[ABCD  1  5 17]  [DABC  6  9 18]  [CDAB 11 14 19]  [BCDA  0 20 20]
[ABCD  5  5 21]  [DABC 10  9 22]  [CDAB 15 14 23]  [BCDA  4 20 24]
[ABCD  9  5 25]  [DABC 14  9 26]  [CDAB  3 14 27]  [BCDA  8 20 28]
[ABCD 13  5 29]  [DABC  2  9 30]  [CDAB  7 14 31]  [BCDA 12 20 32]

/* Round 3. */
/* Let [abcd k s t] denote the operation
     a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
/* Do the following 16 operations. */
[ABCD  5  4 33]  [DABC  8 11 34]  [CDAB 11 16 35]  [BCDA 14 23 36]
[ABCD  1  4 37]  [DABC  4 11 38]  [CDAB  7 16 39]  [BCDA 10 23 40]
[ABCD 13  4 41]  [DABC  0 11 42]  [CDAB  3 16 43]  [BCDA  6 23 44]
[ABCD  9  4 45]  [DABC 12 11 46]  [CDAB 15 16 47]  [BCDA  2 23 48]
```

MD5 Sum

```
/* Round 4. */
/* Let [abcd k s t] denote the operation
     a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
/* Do the following 16 operations. */
[ABCD  0  6 49]  [DABC  7 10 50]  [CDAB 14 15 51]  [BCDA  5 21 52]
[ABCD 12  6 53]  [DABC  3 10 54]  [CDAB 10 15 55]  [BCDA  1 21 56]
[ABCD  8  6 57]  [DABC 15 10 58]  [CDAB  6 15 59]  [BCDA 13 21 60]
[ABCD  4  6 61]  [DABC 11 10 62]  [CDAB  2 15 63]  [BCDA  9 21 64]
```

Conjectures:

- 2^{64} operations needed to create two messages with the same MD5 signature
- 2^{128} operations needed to make a new message with the same MD5 signature as another message