Error Detecting Codes

Nanthawat Anancharoenpakorn

May 2023

Graduate Assignment D
NANTHAWAT ANANCHAROENPAKORN - u7540836
The declaration: "I have read the ANU Academic Skills statement regarding collusion. I have not engaged in collusion in relation to this assignment"

Nanthawat Anancharoenpakorn
28 May 2023

Nowadays, most of the communication channels we use are unreliable due to potential interference. This issue encompasses not only basic channels like telephones but also advanced tools such as satellites. In order to enhance communication reliability, it is crucial to address this problem effectively. One possible solution is to attach one more piece of information to the message to verify if (1) the message has been corrupted and (2) there is no evidence indicating that the message has been corrupted.

1 Error Detecting Codes

Generally, most error detecting codes work as follows. Let A represent the set of all possible messages. Let C denote nonempty set. Let $h : A \to C$ be a function called a *hash function.* When sender want to transmit a message m where $m \in A$, they instead transmit a pair (m, c) with $h(m) = c$ where $(m, c) \in A \times C$. Let (m', c') denote as a pair that recipient will receive. After transmitted, there are two possible cases:

If it went well, then $m = m'$ and $c' = h(m)$ However, it is not possible for recipient to know m for verifying the message. Recipient can only check by compute $h(m')$ against c' If $h(m') \neq c'$ then recipient knows that something went wrong. In the case of $h(m') = c'$, no error is detected. However, it does not mean that $m = m'$. Instead, it come down into three possible cases:

(1) $m = m'$ and $c = c'$ The message was transmitted accurately.

For second and third case, message contains errors but the recipient do not have any evident to detect this.

(2) $m \neq m'$ and $c = c'$ and $h(m) = h(m')$

(3) $m \neq m'$ and $c \neq c'$ and $h(m') = c'$

So we need to minimize the possibilities of both cases. The probability of second case depends on how many different message maps to the same hash-code. Also, the probability of third case depends on how many different message maps to the same hash-code, and how random-like the function h is. Therefore, the decision we make on h can greatly affect the possibilities for both cases.

2 TrySBN-16

An ISBN-16 is a 16 digit code assigned to a publication such as books. The first 15 characters of an TrySBN-16 code are digits that uniquely identify the publication such as the tile, author, edition etc. The sixteenth

character is either a digit or the letter X, which is used to detect errors in transmission or recording.

Formally, let $X = 16$ and

$$\begin{aligned} D &= \{0, 1, 2, \ldots, 9\}, \\ C &= \{0, 1, 2, 3, ..., 14, 15, X\}, \\ A &= D \times D \times \cdots \times D \text{ (15 times)} \end{aligned}$$

Let define a function $h : A \to C$ by the rule

$$\forall (a_1, ..., a_{15}) \in A \quad h((a_1, ..., a_{15})) \equiv \sum_{i=1}^{15} i a_i \pmod{17}$$

For the rest of this document, $a \equiv b$ means a and b are congruent modulo 17 or $a \equiv b \pmod{17}$. When the TrySBN-16 is written, no parentheses or commas are included. For instance, we have the message $m = (1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5)$ and $h(m) = 3$. The transmission is the message followed by the hash-code. In this case, the finalized message will be $m' = 1234567890123453$

Definition 1.

An element $(a_1, ..., a_{15}, c) \in D \times \cdots \times D \times C$ is a *valid* TrySBN-16 if

$$c \equiv \sum_{i=1}^{15} i a_i \pmod{17}$$

Lemma 2.

An element $(a_1, ..., a_{15}, a_{16}) \in D \times \cdots \times D \times C$ is a *valid* TrySBN-16 if and only if

$$\sum_{i=1}^{16} i a_i \equiv 0 \pmod{17}$$

Proof. Let $(a_1, ..., a_{15}, a_{16}) \in D \times \cdots \times D \times C$ Then

$$
\begin{aligned}
&(a_1, ..., a_{15}, a_{16}) \qquad \text{(is a valid TrySBN-16)} \\
&\Leftrightarrow a_{16} \equiv \sum_{i=1}^{15} i a_i \qquad \text{(using the definition above)} \\
&\Leftrightarrow 0 \equiv (\sum_{i=1}^{15} i a_i) - (a_{16}) \\
&\Leftrightarrow 0 \equiv (\sum_{i=1}^{15} i a_i) + (-1) a_{16} \\
&\Leftrightarrow 0 \equiv (\sum_{i=1}^{15} i a_i) + 16 a_{16} \qquad \text{(because } 16 \equiv -1\text{)} \\
&\Leftrightarrow 0 \equiv (\sum_{i=1}^{16} i a_i)
\end{aligned}
$$

Theorem 3

The TrySBN-16 detects any transmission error where exactly one character is changed. Let $m = (a_1, \ldots a_{15}, a_{16})$ and $m' = (a'_1, \ldots, a'_{15}, a'_{16})$ be elements of $D \times \cdots \times D \times C$. If m is a valid TrySBN-16, and m and m' differ in exactly one component, then m' is not a valid TrySBN-16.

Proof. Suppose that m is a valid TrySBN-16, and m and m' differ in exactly one component. Then there exists an integer j such that $1 \leq j \leq 16$ and $a'_j \neq a_j$ and $a'_k \neq a_k$ for all integers k such that $1 \leq k \leq 16$ and $k \neq j$. By Lemma 2, to show that m' is not valid TrySBN-16 it suffices to show that

$$0 \not\equiv \sum_{i=1}^{16} i a'_i$$

As we know that $a \equiv b \pmod{d}$ if and only if d divides $b - a$ Thus it suffices to show that

$$\sum_{i=1}^{16} i a'_i$$

is not divisible by 17 Now

$$
\begin{aligned}
\sum_{i=1}^{16} i a'_i &\equiv \sum_{i=1}^{16} i a'_i - 0 \\
&\equiv \sum_{i=1}^{16} i a'_i - \sum_{i=1}^{16} i a_i \qquad \text{(by lemma, because m is a valid TrySBN-16)} \\
&\equiv \sum_{i=1}^{16} i (a'_i - a_i) \qquad \text{(using algebraic properties of summation)} \\
&\equiv j(a'_j - a_j) \qquad \text{(because } a_k = a'_k \text{when } k \neq j\text{)}
\end{aligned}
$$

Since $1 \leq j \leq 16$, 17 does not divide j Since $0 \leq a_j \leq 16$ and $0 \leq a'_j \leq 16$, $-16 \leq a_j - a'_j \leq 16$. Since $a_j \neq a'_j, a_j - a'_j \neq 0$ Hence

$$a_j - a'_j \in \{-16, \ldots, -1\} \cup \{1, \ldots, 16\},$$

and 17 does not divide $a_j - a'_j$. By the Prime Divisibility Property, 17 does not divide $j(a_j - a'_j)$. Hence m' is not a valid TrySBN-16.

Theorem 4

If any two distance digits in a valid TrySBN-16 are transposed, then the resulting string is not a valid TrySBN-16.

Proof. Let $(a_1, ..., a_{16}), (b_1, ..., b_{16}) \in D \times \ldots \times D \times C$. Suppose that $(a_1, ..., a_{16})$ is a valid TrySBN-16 and $(a'_1, ..., a'_{16})$ is obtained from $(a_1, ..., a_{16})$ by transposing two distinct digits. Then there exist integers j, k such that $1 \leq j < k \leq 16$ and $a'_j = a_k$ and $a'_k = a_j$ and $a'_i = a_i$ for all integers i such that $1 \leq i \leq 16$ and $i \neq j, k$. The fact that the digits transposed are distinct means that $a_j \neq a_k$. By Lemma 2, to show that $(a'_1, ..., a'_{16})$ is not a valid TrySBN-16 it suffices to show that

$$0 \not\equiv \sum_{i=1}^{16} i a'_i$$

As we know that $a \equiv b \pmod{d}$ if and only if d divides $b - a$ Thus it suffices to show that

$$\sum_{i=1}^{16} i a'_i$$

is not divisible by 17. Now

$$\begin{aligned}
\sum_{i=1}^{16} i a'_i &\equiv \sum_{i=1}^{16} i a'_i - 0 \\
&\equiv \sum_{i=1}^{16} i a'_i - \sum_{i=1}^{16} i a_i \qquad \text{(by lemma, because m is a valid TrySBN-16)} \\
&\equiv \sum_{i=1}^{16} i(a'_i - a_i) \qquad \text{(using algebraic properties of summation)} \\
&\equiv j(a'_j - a_j) + k(a'_k - a_k) \qquad \text{(because } a_i = a'_i \text{ when } i \neq j, k) \\
&\equiv j(a_k - a_j) + k(a_j - a_k) \qquad \text{(because } a'_j = a_k \text{ and } a'_k = a_j) \\
&\equiv (k - j)(a_j - a_k)
\end{aligned}$$

Since $1 \leq j < k \leq 16$, we have that $1 \leq k - j \leq 15$; it follows that 17 does not divide $k - j$. Since $0 \leq a_j, a_k \leq 16$ and $a_j \neq a_k$, we have that $-16 \leq a_j - a_k \leq 16$ and $a_j - a_k \neq 0$; it follows that 17 does not divide $a_j - a_k$. By the Prime Divisibility Property, 17 does not divide $(k - j)(a_j - a_k)$. Hence $\sum_{i=1}^{16} i a'_i$ is not divisible by 17.