链滴

最初的谜题

作者：plus7wist
原文链接：https://ld246.com/article/1591107606795
来源网站：链滴

谜题系列

以前玩过一些谜题，说实话，我水平不高。但就像说葡萄酸的狐狸，我的品味也有所不同。这就让我生了创作一个系列谜题的想法，于是这就是第一题了。

这道题目让我回忆起我的高中时代，那时候的我跟现在真的是差别太大了。

题目

310441266559681555559

加盐哈系

这一系列谜题都不会延迟公布答案，这是为了防止猜到答案的朋友们需要挨过不友好的等待。但也不完全直接公布，那样不免会影响猜谜体验。所以我会公布答案的校验和，这样猜出答案的人就可以确自己答案是否正确了。

我选择这样的方式计算哈系：

$$\mathfrak{R} = \operatorname{sha256}(\operatorname{sha256}(\mathfrak{P}) + \mathfrak{S})$$

$\mathfrak{P}$ 是谜题的答案；$\frak{S}$ 是一个简单的单词，用作加盐；$\mathfrak{R}$ 是最终的哈系可以用一个 bash 脚本完成上面的过程：

```
#! /usr/bin/env bash

PROGRAM=$(basename "$0")

show_help() {
    cat <<-EOF
```

原文链接：最初的谜题

```
    name:
      $PROGRAM - sha256(sha256(INPUT) + SALT)
    usage:
      $PROGRAM salt-hash.sh -s SALT -i INPUT
      $PROGRAM -h
    options:
      -s SALT   specify a salt string.
      -i INPUT  input secret string. '-' means read from stdin.
      -h        show this help and exit.
    EOF
}

fail() {
    echo "ERROR: $*"
    exit 1
}

declare OPT_SALT OPT_INPUT

while getopts 'i:s:h' opt; do
    case $opt in
        h) show_help; exit;;
        i) OPT_INPUT="$OPTARG";;
        s) OPT_SLAT="$OPTARG";;
        ?) exit 1;;
    esac
done

[ -z "$OPT_SALT" ] && fail "need option -s"
[ -z "$OPT_INPUT" ] && fail "need option -i"

run() {
    local tmp
    if [ "$OPT_INPUT" == "-" ]; then
        tmp="$(sha256sum | cut -d' ' -f1)"  # read from stdin
    else
        tmp="$(echo -n "$OPT_INPUT" | sha256sum | cut -d' ' -f1)"
    fi
    echo -n "$tmp$OPT_SALT" | sha256sum | cut -d' ' -f1
}

run
```

最终我会按照这样的格式写出答案：

- \mathfrak{S}
- \mathfrak{R}

假设某题的答案是 apple，而我给出的盐是 puzzle，那么运行上述脚本：

```
$ ./salt-hash.sh -s puzzle -i apple
933604f1d196b04dcab78347f74a66480829c3b2759422a671d1914e4cca3d29
```

于是我会声称答案是：

- puzzle
- 933604f1d196b04dcab78347f74a66480829c3b2759422a671d1914e4cca3d29

在线工具

我给不方便自己做哈系的同学，写了一个小工具。

https://plus7wist.gitee.io/puzzle-salt-hash

本题答案

- puzzle
- 5a363bf4b2a7db6483a0068642a6eb6ecb0f539ef57d406ad718ac984b80aefe

发布答案

如果你猜出了答案，请不要直接写出来，而是也按照上述过程，自选一个 \mathfrak{S}，写出一个哈。这样一来，知道答案的人就可以确认你的答案是正确的，并且由衷地赞美你 ada。

艰难

题目修订的第一版本，我直接公布了答案的校验和，但因为答案是个常用单词，所以有几位 HacPaier 直接查了彩虹表……