Hochschule Konstanz
Department of Computer Science

Cracking Passwords with C++

MSI - DIMA - Task 2

Nadina Brodt, Johannes Brandenburger, Tobias Tögel

github.com/johannesbrandenburger/htwg-msi-dima

Number of Possible Passwords for Each System

```
System 1 ([a-z][A-Z][0-9],    5 digits):
                                                916,132,832
                                        nine hundred million

System 2 ([a-z][A-Z][0-9],   10 digits):
                                      839,299,365,868,340,224
                                    eight hundred quadrillion

System 3 ([a-z][A-Z][0-9], 5-10 digits):
                                      853,058,371,851,163,296
                                    eight hundred quadrillion
```

Approach to Crack the Passwords

- C++
- 2 main functions
  - to get number of different passwords
  - to get password for each number

```cpp
long long int count_combinations(
    string alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",
    vector<int> allowed_lengthes = {5, 6, 7, 8, 9, 10}
) { … }

string get_code_from_number_with_variable_length(
    long long int number,
    string alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789",
    vector<long long int> allowed_lengthes = {5, 6, 7, 8, 9, 10}
) { … }
```

Approach to Crack the Passwords

- wrapper function to „easily“ crack a system with a given hash

```
// crack system 1
string result_system_1 = try_crack_hash(
    searched_hash_system_1,
    „abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789“, // alphabet
    vector<long long int>{5}, // 5 digits long
    false                     // no debug output
);
cout << "result: " << result_system_1 << endl;

// output:
// try to crack system 1 (5 signs) with SHA-1: 7738d1909d7dee18196f733d0d508d871d05cc80
// current code: WmI42 | progress: 88.5679% | remaining minutes to test all combinations: 1.96627
// took 15min to crack the hash
// result: MsI42
```

System 2 and 3

- system 2 and 3 not crackable with this approach due to time constraint
- would take around 28 thousand years on a MacBook with M1
- this brute force approach with indices is not very efficient
  - more efficient: nested for-loops (13 thousand years on same device)
  - but not as flexible

Hochschule Konstanz
Department of Computer Science

Cracking Passwords with C++

MSI - DIMA - Task 2

Nadina Brodt, Johannes Brandenburger, Tobias Tögel

github.com/johannesbrandenburger/htwg-msi-dima