Hashing Slides

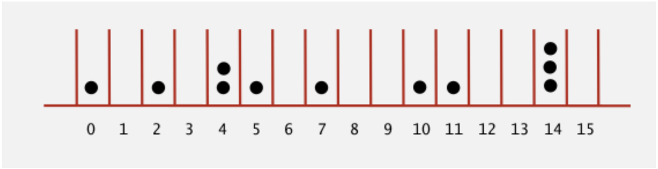
Chris Tralie

Duke University, ECE / Math

12/5/2018

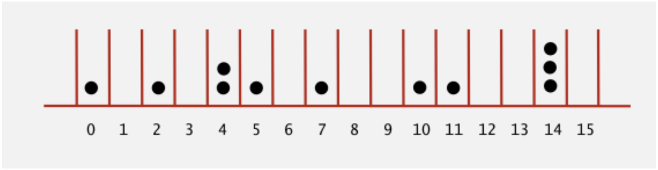
Theory: Pseudorandom Functions

- Each key equally likely to map to integer between 0 and $M-1$
- Throw balls blindfolded into bins



Theory: Pseudorandom Functions

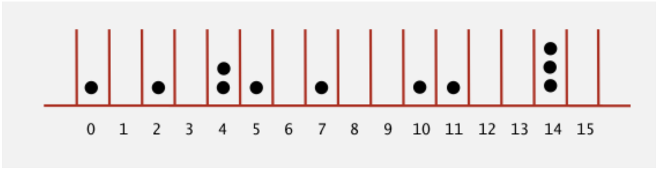
- Each key equally likely to map to integer between 0 and $M-1$
- Throw balls blindfolded into bins



- Birthday paradox: Expect collision after $\approx \sqrt{\pi M/2}$ tosses

Theory: Pseudorandom Functions

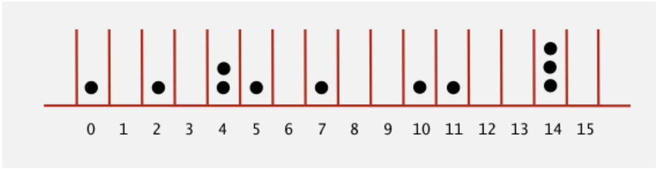
- Each key equally likely to map to integer between 0 and $M-1$
- Throw balls blindfolded into bins



- Birthday paradox: Expect collision after $\approx \sqrt{\pi M/2}$ tosses
- Expect every bin has ≥ 1 ball after $\approx M \ln M$ tosses

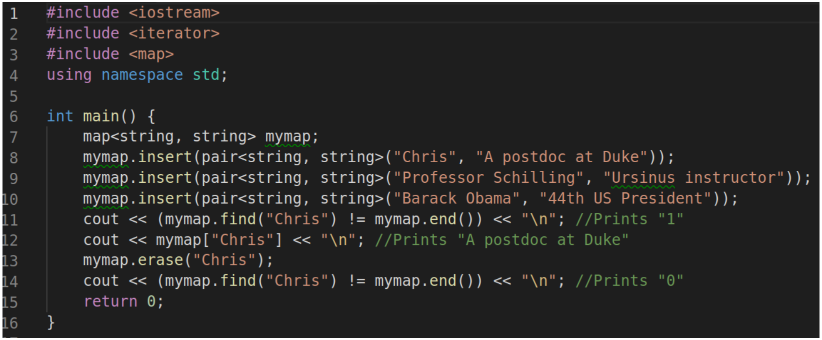
Theory: Pseudorandom Functions

- Each key equally likely to map to integer between 0 and $M-1$
- Throw balls blindfolded into bins



- Birthday paradox: Expect collision after $\approx \sqrt{\pi M/2}$ tosses
- Expect every bin has ≥ 1 ball after $\approx M \ln M$ tosses
- After M tosses, expected most loaded bin has $\Theta(\log M / \log \log M)$ balls

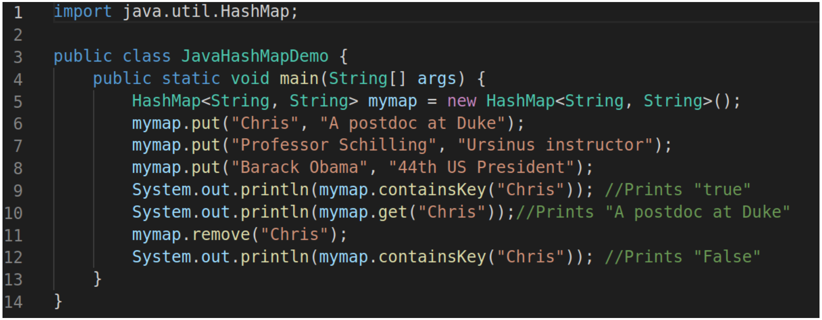
C++ Implementation



```cpp
#include <iostream>
#include <iterator>
#include <map>
using namespace std;

int main() {
    map<string, string> mymap;
    mymap.insert(pair<string, string>("Chris", "A postdoc at Duke"));
    mymap.insert(pair<string, string>("Professor Schilling", "Ursinus instructor"));
    mymap.insert(pair<string, string>("Barack Obama", "44th US President"));
    cout << (mymap.find("Chris") != mymap.end()) << "\n"; //Prints "1"
    cout << mymap["Chris"] << "\n"; //Prints "A postdoc at Duke"
    mymap.erase("Chris");
    cout << (mymap.find("Chris") != mymap.end()) << "\n"; //Prints "0"
    return 0;
}
```

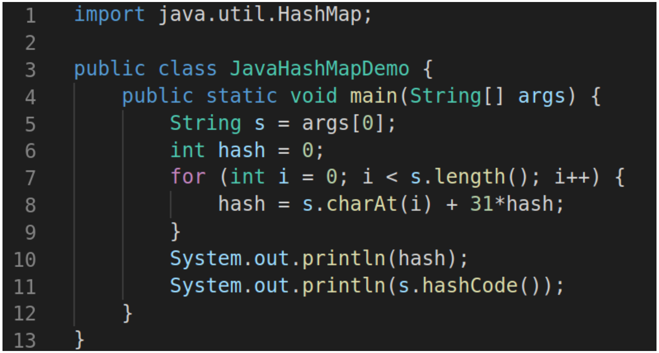
Java Implementation



```java
import java.util.HashMap;

public class JavaHashMapDemo {
    public static void main(String[] args) {
        HashMap<String, String> mymap = new HashMap<String, String>();
        mymap.put("Chris", "A postdoc at Duke");
        mymap.put("Professor Schilling", "Ursinus instructor");
        mymap.put("Barack Obama", "44th US President");
        System.out.println(mymap.containsKey("Chris")); //Prints "true"
        System.out.println(mymap.get("Chris"));//Prints "A postdoc at Duke"
        mymap.remove("Chris");
        System.out.println(mymap.containsKey("Chris")); //Prints "False"
    }
}
```

Java’s scheme



```java
import java.util.HashMap;

public class JavaHashMapDemo {
    public static void main(String[] args) {
        String s = args[0];
        int hash = 0;
        for (int i = 0; i < s.length(); i++) {
            hash = s.charAt(i) + 31*hash;
        }
        System.out.println(hash);
        System.out.println(s.hashCode());
    }
}
```

Hello $\rightarrow$ 69609650

Chris $\rightarrow$ 65087095

Ursinus $\rightarrow$ 1501567193

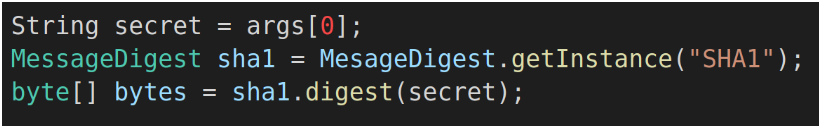
Adversarial Attack?

Adversarial Attack?

More stable: MD4, MD5, SHA-0, SHA-1, SHA-2, WHIRLPOOL, RIPEMD-160

Adversarial Attack?

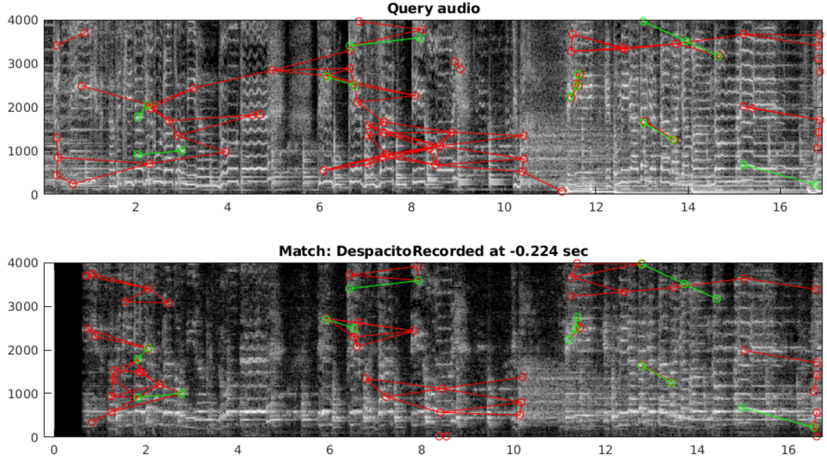
More stable: MD4, MD5, SHA-0, SHA-1, SHA-2, WHIRLPOOL, RIPEMD-160



```
String secret = args[0];
MessageDigest sha1 = MesageDigest.getInstance("SHA1");
byte[] bytes = sha1.digest(secret);
```

Applications?

The Shazam Technique



▷ Audio fingerprinting works well for exact recordings, possibly degraded

Thank You!

Contact: chris.tralie@gmail.com