Team '; drop table *; --;-), PlaidCTF 2014, challenge “g++”

Given was the following task:

> Although it seems like The Plague's projects are open source, it's not quite so simple to figure out what the source code does. We believe this project is supposed to print out secret information, but the KEY variable in the Makefile has been lost. Find the key, build the project, get us the information.

Together with an obfuscated piece of code, which incorporates the flag as a key.
Since the code itself is heavily obscured by #define directives, it makes sense to let the compiler first replace them. We do this with the command **g++ -E -o solveme2.cpp solveme.cpp**.

With the new result, we can now start our analysis. What follows is an excerpt of the first interesting part of the code:

```
template <int n>
struct gg {
  static const int r =
(m<((u<n,n|2>::r)),(((1<<(1<<3))|1)),(((1<<(1<<3))|1))>::r)
                                  - r<(n>>2),((n)&3)>::rr;
};

template <int n>
struct vvv {
  static const int r = gg<n>::r|gg<n+1>::r|gg<n+2>::r|gg<n+3>::r;
};

template <int n>
struct vv {
  static const int r = vvv<0>::r|vvv<4>::r|vvv<8>::r|vvv<12>::r;
};

int main() {
  if (!vv<0>::r) {
      //[...]
  }
  else {
      std::cout << "Wrong\n";
  }
  return 0;
}
```

We notice that *vv<0>* has to be zero if the key bytes are correct. If we follow the tree upwards, it becomes clear that *gg<t>* has to be zero for *0<=t<=15*. As it turns out, the first part of the equation in gg is constant in respect to t, while only *r<...>::rr* depends on the key bytes. With this in mind we can reduce the problem to *r<...>::rr* = *x* for a fixed x we can readily calculate.

By manually analysing the structures, we are able to further reduce the problem to solving the structure d, and to reduce the structure d to the following code:

```
template <int s, int t, int u, int v, int w, int x, int y, int z>
struct d {
  static const int r = (s*w+t*x+u*y+v*z) % 257;
}
```

Where s,t,u,v are static calculable values, and w,x,y,z are four key bytes. This structure gets called 16 times, each time with another value combination. Knowing the values s,t,u,v and the expected result, we can formulate the problem as 4 linear equation systems modulo 257 and solve them for the key bytes.

The result is:
C++_m0re_lyk_C--