

A.V.ISOGENIES, a library for computing isogenies between abelian varieties

Gaetan BISSON Romain COSSET Damien ROBERT

CARAMEL group
LORIA, Nancy, France

s/CACAO/CARAMEL/g



Who are these people?

s/CACAO/CARAMEL/g



Who are these people?

SERIOUS PEOPLE

Jérémie D.
Pierrick G.
Emmanuel T.
Paul Z.

→ ECC'11

s/CACAO/CARAMEL/g



Who are these people?

SERIOUS PEOPLE

Jérémie D.
Pierrick G.
Emmanuel T.
Paul Z.

→ ECC'11

YOUNG GUYS

Gaetan B.
Romain C.
Nicolas E.
Damien R.

s/CACAO/CARAMEL/g



Who are these people?

SERIOUS PEOPLE

Jérémie D.
Pierrick G.
Emmanuel T.
Paul Z.

→ ECC'11

YOUNG GUYS

Gaetan B.
Romain C.
~~Nicolas E.~~
Damien R.

The way *they* code

```
d[5],Q[999
(;i--;e=scanf("%"
++i<A          ;++Q[
R:i);          for(;i
--;N           +=!M*Q
+E*E-         R*L*
E=i,L=M,a=4;a;C=
%A,E=C%A+a
```

```
]={0};main(N
"d",d+i));for(A
i*i%          A),R=
--;)          for(M
[E%A          ],e+=
L%A)         %A))
i*E+R*M*L,L=(M*E
--[d]);printf
```

```
C,A,
R,a,
M,E,
L,i=
5,e,
){for
=*d;
i[Q]?
=A;M
Q[(A
for(
+i*L)
("%d"
"\n",
(e+N*
N)/2
-A);}]
```

The way *they* code

```
d[5],Q[999
(;i--;e=scanf("%"
++i<A          ;++Q[
R:i);          for(;i
--;N           +=!M*Q
+E*E-         R*L*
E=i,L=M,a=4;a=C=
%A,E=C%A+a
```

```
]={0};main(N
"d",d+i));for(A
i*i%          A),R=
--;)          for(M
[E%A          ],e+=
L%A)          %A))
i+E+R*M*L,L=(M*E
--[d]);printf
```

```
C,A,
R,a,
M,E,
L,i=
5,e,
){for
=*d;
i[Q]?
=A;M
Q[(A
for(
+i*L)
("%d"
"\n",
(e+N*
N)/2
-A);};
```

≈ 250 glyphs

The way *they* code

counts points on Jacobians of genus-two curves

```
d[5],Q[999
(;i--;e=scanf("%"
++i<A          ;++Q[
R:i);          for(;i
--;N           +=!M*Q
+E*E-         R*L*
E=i,L=M,a=4;a;C=
%A,E=C%A+a
```

```
]={0};main(N
"d",d+i));for(A
i*i%          A),R=
--;)          for(M
[E%A          ],e+=
L%A)          %A))
i*E+R*M*L,L=(M*E
--[d]);printf
```

```
C,A,
R,a,
M,E,
L,i=
5,e,
){for
=*d;
i[Q]?
=A;M
Q[(A
for(
+i*L)
("%d"
"\n",
(e+N*
N)/2
-A);};
```

≈ 250 glyphs

The way *they* code

counts points on Jacobians of genus-two curves

```
d[5],Q[999
(;i--;e=scanf("%"
++i<A          ;++Q[
R:i);          for(;i
--;N           +=!M*Q
+E*E-         R*L*
E=i,L=M,a=4;a=C=
%A,E=C%A+a
```

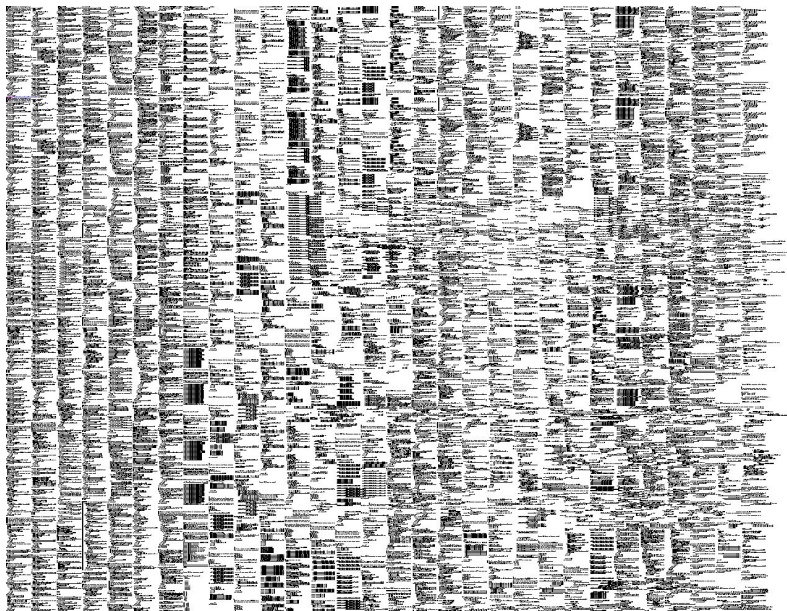
```
]={0};main(N
"d",d+i));for(A
i*i%          A),R=
--;)          for(M
[E%A          ],e+=
L%A)         %A))
i+E+R*M*L,L=(M*E
--[d]);printf
```

```
C,A,
R,a,
M,E,
L,i=
5,e,
){for
=*d;
i[Q]?
=A;M
Q[(A
for(
+i*L)
("%d"
"\n",
(e+N*
N)/2
-A);};
```


≈ 250 glyphs

reasonably fast

The way *we* code



The way *we* code



not even a complete $g \geq 2$ SEA implementation

The way *we* code

reasonably slow

not even a complete $g \geq 2$ SEA implementation

The way *we* code

reasonably slow

various subtly confusing structures

not even a complete $g \geq 2$ SEA implementation

The way *we* code

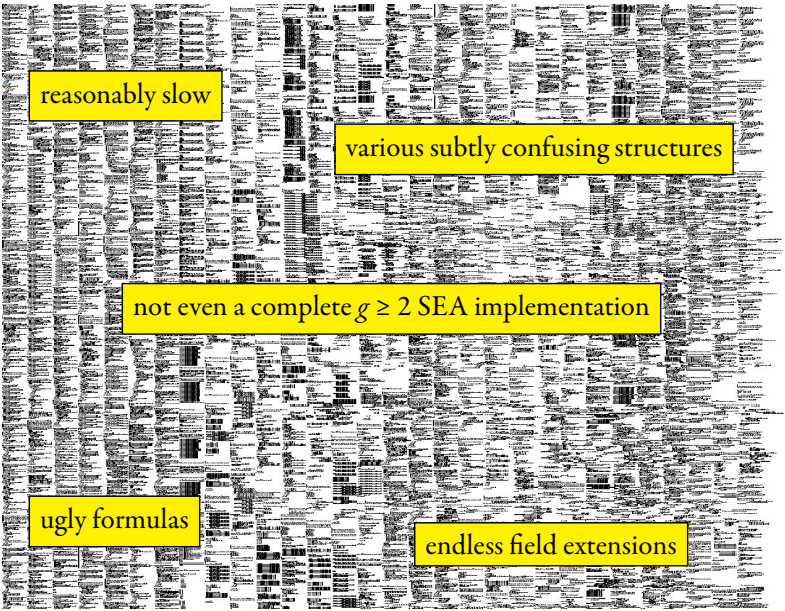
reasonably slow

various subtly confusing structures

not even a complete $g \geq 2$ SEA implementation

ugly formulas

The way *we* code

The background of the slide is a dense, black and white image of code, likely from a cryptographic implementation. The code is arranged in vertical columns, with some lines indented. The overall appearance is that of a complex, multi-line program. Five yellow callout boxes with black text are overlaid on the code, pointing to various parts of the implementation. The boxes contain the following text: 'reasonably slow', 'various subtly confusing structures', 'not even a complete $g \geq 2$ SEA implementation', 'ugly formulas', and 'endless field extensions'.

reasonably slow

various subtly confusing structures

not even a complete $g \geq 2$ SEA implementation

ugly formulas

endless field extensions

The way *we* code

reasonably slow

various subtly confusing structures

not even a complete $g \geq 2$ SEA implementation

redundant chunks of code

ugly formulas

endless field extensions

The way *we* code



reasonably slow

various subtly confusing structures

not even a complete $g \geq 2$ SEA implementation

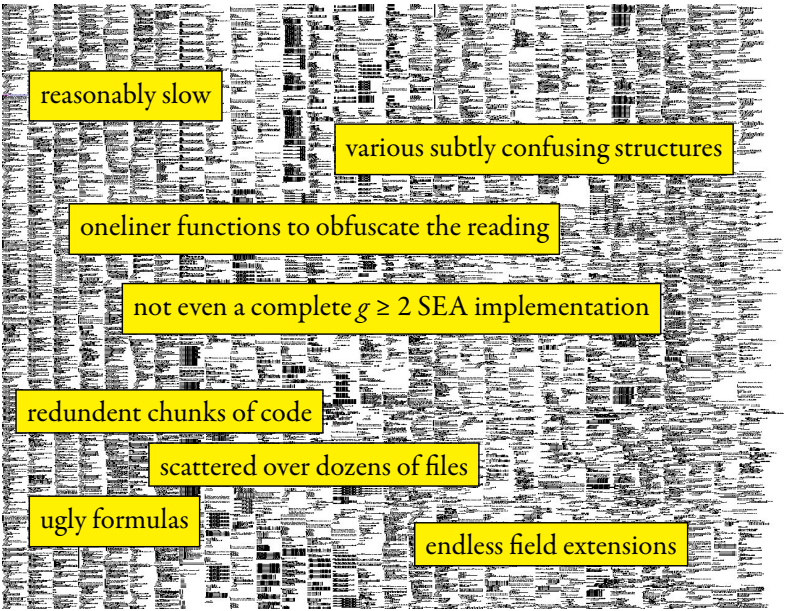
redundant chunks of code

scattered over dozens of files

ugly formulas

endless field extensions

The way *we* code



reasonably slow

various subtly confusing structures

oneliner functions to obfuscate the reading

not even a complete $g \geq 2$ SEA implementation

redundent chunks of code

scattered over dozens of files

ugly formulas

endless field extensions

The way *we* code

reasonably slow

randomly appearing FrenGLISH comments

various subtly confusing structures

oneliner functions to obfuscate the reading

not even a complete $g \geq 2$ SEA implementation

redundent chunks of code

scattered over dozens of files

ugly formulas

endless field extensions

The way *we* code

reasonably slow

randomly appearing FrenGLISH comments

various subtly confusing structures

oneliner functions to obfuscate the reading

not even a complete $g \geq 2$ SEA implementation

redundent chunks of code

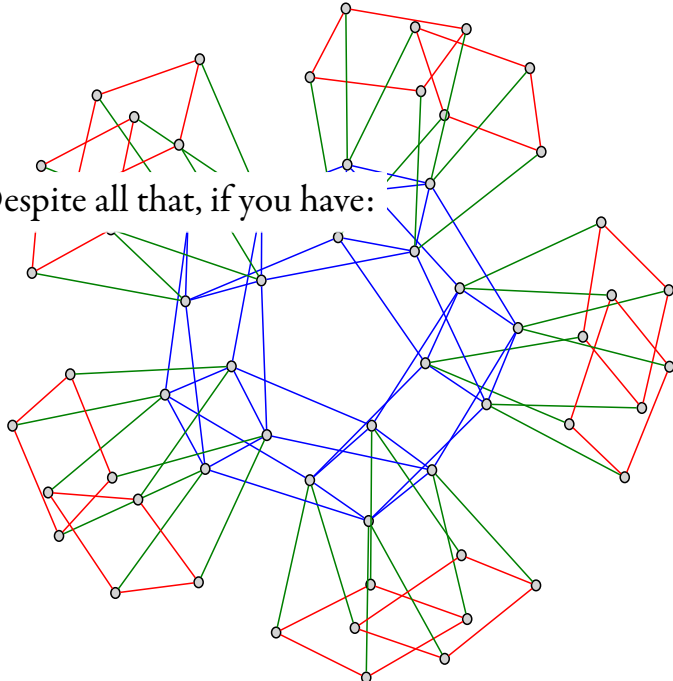
BONUS: undocumented bugs

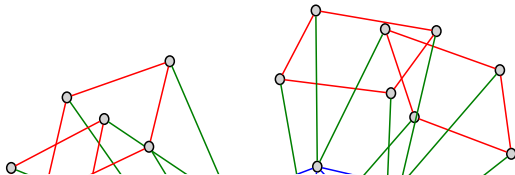
scattered over dozens of files

ugly formulas

endless field extensions

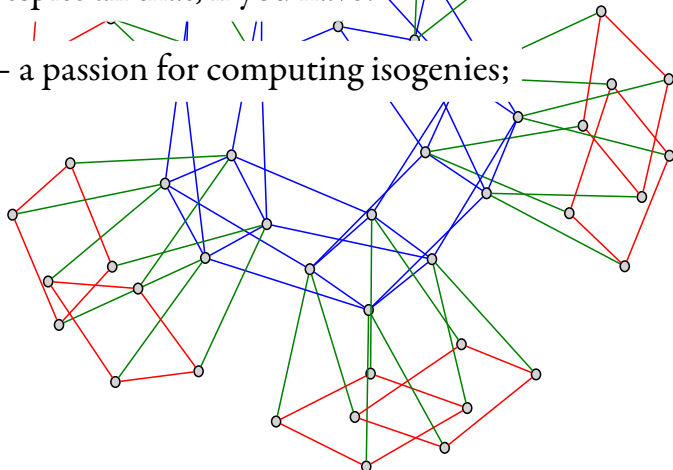
Despite all that, if you have:

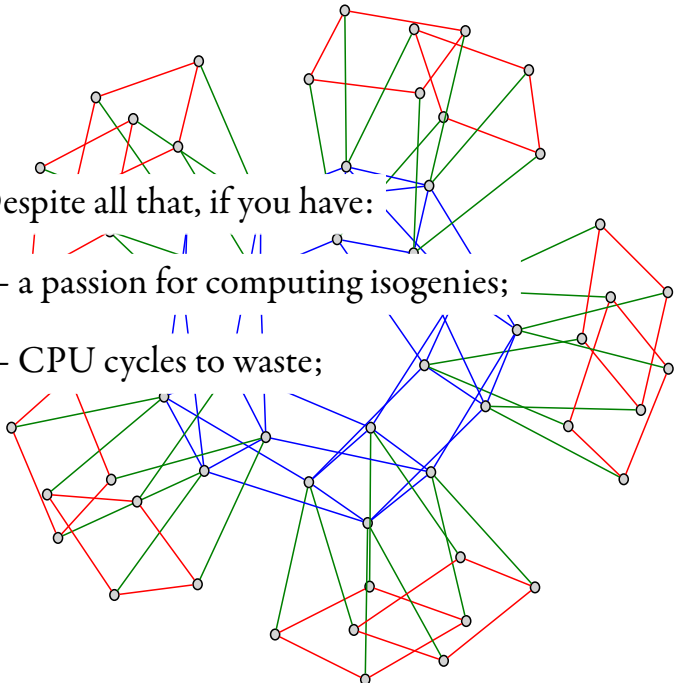




Despite all that, if you have:

— a passion for computing isogenies;





Despite all that, if you have:

— a passion for computing isogenies;

— CPU cycles to waste;



Despite all that, if you have:

— a passion for computing isogenies;

— CPU cycles to waste;

drop us an email and get an LGPL'ed library for free!