We create a polynomial and find that it is squarefee modulo all primes > 7.

```
gap> x:=X(Rationals,"x");;f:=x^8-16*x^4-98;;
gap> Set(Factors(Discriminant(f)));
[ -2, 2, 3, 7 ]
gap> l:=Filtered([8..1000],IsPrimeInt);;Length(l);
164
```

Lets consider the first such prime, 11. We reduce f modulo 11 and find that it is the product of two factors of degree 4.

```
gap> p:=l[1];
11
gap> UnivariatePolynomial(GF(p),CoefficientsOfUnivariatePolynomial(f)
>                          *One(GF(p)),1);
x_1^8+Z(11)^9*x_1^4+Z(11)^0
gap> fmod:=UnivariatePolynomial(GF(p),CoefficientsOfUnivariatePolynomial(f)
>                                *One(GF(p)),1);
x_1^8+Z(11)^9*x_1^4+Z(11)^0
gap> List(Factors(fmod),Degree);
[ 4, 4 ]
```

The 'Collected' command transforms such a list into a nicer form: 2 factors of degree 4. We start a list in which we collect this information and do the same calculation in a loop over all other primes.

```
gap> shape:=Collected(List(Factors(fmod),Degree));
[ [ 4, 2 ] ]
gap> a:=[];;
gap> Add(a,shape);

gap> for i in [2..Length(l)] do
>   p:=l[i];
>   fmod:=UnivariatePolynomial(GF(p),CoefficientsOfUnivariatePolynomial(f)
>   *One(GF(p)),1);
>   shape:=Collected(List(Factors(fmod),Degree)); Add(a,shape);
> od;
```

We now collect the information over all primes, considering (inverse) frequencies out of the 164 primes in $[7, 1000]$. For example 1/3 of all primes gave a factorization into 2 linear factors and 3 quadratic, 1/54 of all primes into a product of 8 linear factors.

```
gap> Collected(a);
[ [[[1,2],[2,3]],42], [[[1,4],[2,2]],9], [[[1,8]],3],
  [[[2,4]],23],[[[4,2]],45], [[[8,1]],42] ]
```

```
gap>List(Collected(a),i->[i[1],Int(164/i[2])]);
[ [[[1,2],[2,3]],3], [[[1,4],[2,2]],18], [[[1,8]],54],[[[2,4]],7],
  [[[4,2]],3], [[[8,1]],3] ]
```

We now want to compare this information to the cycle shape distribution in a permutation group.

For this we use an existing classification of transitive subgroups of S_n up to conjugacy. (Such lists have been computed up to degree 35 at the time of writing.)

Consider for example the 10-th group in the list of transitive subgroups degree 8. We collect the cycle structures of all elements and again count frequency information: 1/16 of all elements have only 1-cycles, 1/2 have two 4-cycles, 1/8 has two 2-cycles, 1/3 four 2-cycles.

```
gap> g:=TransitiveGroup(8,10);
[2^2]4
gap> Collected(List(Elements(g),CycleStructurePerm));
[ [ [  ], 1 ], [ [ ,, 2 ], 8 ], [ [ 2 ], 2 ], [ [ 4 ], 5 ] ]
gap> List(Collected(List(Elements(g),CycleStructurePerm)),
>         i->[i[1],Int(Size(g)/i[2])]);
[ [ [  ], 16 ], [ [ ,, 2 ], 2 ], [ [ 2 ], 8 ], [ [ 4 ], 3 ] ]
```

This apparently does not agree with the frequencies we got. Thus do this calculation for all (50) transitive groups of degree 8:

```
gap> NrTransitiveGroups(8);
50
gap> e:=[];;
gap> for i in [1..50] do
> g:=TransitiveGroup(8,i);
> freq:=List(Collected(List(Elements(g),CycleStructurePerm)),
> i->[i[1],Int(Size(g)/i[2])]);
> Add(e,freq);
> od;
```

We certainly are only interested in groups which contain cycle shapes as we observed. We thus check, which groups (given by indices) contain elements that are: three 2-cycles, two 2-cycles, four 2-cycles, two 4-cycles, and one 8-cycle.

```
gap> sel:=[1..50];;
gap> sel:=Filtered(sel,i->ForAny(e[i],j->j[1]=[3]));
[ 6, 8, 15, 23, 26, 27, 30, 31, 35, 38, 40, 43, 44, 47, 50 ]
gap> sel:=Filtered(sel,i->ForAny(e[i],j->j[1]=[2]));
[ 15, 26, 27, 30, 31, 35, 38, 40, 44, 47, 50 ]
gap> sel:=Filtered(sel,i->ForAny(e[i],j->j[1]=[4]));
[ 15, 26, 27, 30, 31, 35, 38, 40, 44, 47, 50 ]
gap> sel:=Filtered(sel,i->ForAny(e[i],j->j[1]=[,,2]));
[ 15, 26, 27, 30, 31, 35, 38, 40, 44, 47, 50 ]
```

```
gap> sel:=Filtered(sel,i->ForAny(e[i],j->j[1]=[,,,,,,1]));
[ 15, 26, 27, 35, 40, 44, 47, 50 ]
```

8 Groups remain. All but the first contain elements of shapes we did not observe, but we can also consider frequencies:

```
gap> e{sel};
[ [[[],32],[[,,,,,,1],4],[[,,2],4],[[2],16],[[3],4],[[4],6]],
[...]
   [[2],192],[[2,,1],32],[[2,1],24],[[3],96],[[4],384]]]
```

Comparing with the frequencies in the group again, we find that the first group (index 15) gives the best correspondence overall.

```
gap> List(Collected(a),i->[i[1],Int(164/i[2])]);
[ [[[1,2],[2,3]],3], [[[1,4],[2,2]],18], [[[1,8]],54],[[[2,4]],7],
  [[[4,2]],3], [[[8,1]],3] ]
```

(The group has order 32, it has a structure $(C_2 \times D_8) \rtimes C_2$.)

GAP has a function that does all of this together in one. `ProbabilityShapes` returns a list[1] of the most likely candidates for the Galois group.

```
gap> ProbabilityShapes(f);
[ 15 ]
```

Proving the structure So far the argument has been probabilistic. GAP also can give an exact proof. We turn up the print level to see what it is doing:

```
gap> GaloisType(f);
#I  Partitions Test
#I  cands:[ 15, 26, 27, 35, 40, 44, 47, 50 ]
#I  2-Set Resolvent
#I  trying res nr. 3
#I  Candidates :[ 15, 26, 27, 35 ]
#I  2-Seq Resolvent
#I  Candidates :[ 15, 26, 35 ]
#I  3-Set Resolvent
#I  trying res nr. 3
#I  Candidates :[ 15 ]
15
```

[1]There are groups with exactly the same frequencies

What GAP calls the *k-set resolvent* is the polynomial from Kronecker's construction, specialized at $t_1 = t_2 = \cdots = t_k = 1$, $t_{k+1} = \cdots = t_n = 0$. Its factors correspond to the action of the Galois group $\leq S_n$ on k-sets.

You can construct such a polynomial with a special function (that might do some transformations on x) which uses symmetric function theory. One also could do so with resultants, but that is slower.

```
gap> r:=GaloisSetResolvent(f,3);
#I  trying res nr. 3
x^56+1176/113*x^55+773820/12769*x^54+363171240/1442897*x^53+135037483542/16304\
[...]
0891741374132667448923745388137058865 13*x-312773860274155623997100440295885644\
81/130210891741374132667448923745388137058865 13
gap> List(Factors(r),Degree);
[ 8, 16, 16, 16 ]
```

We see that the Galois group has orbits of length 8 and 16 (thrice) on 3-sets. So it cannot be S_n.

Let us now verify its type. We compute all subgroups between the group claimed and S_n:

```
gap> s:=SymmetricGroup(8);
Sym( [ 1 .. 8 ] )
gap> g:=TransitiveGroup(8,15);
[1/4.cD(4)^2]2
gap> u:=IntermediateSubgroups(s,g);
rec( inclusions := [ [ 0, 1 ], [ 1, 2 ], [ 1, 3 ], [ 2, 4 ], [ 2, 5 ],
      [ 3, 4 ], [ 4, 6 ], [ 5, 6 ] ],
  subgroups := [ Group([ (1,2,3,4,5,6,7,8), (1,5)(3,7), (1,6)(2,5)(3,4)
[...]
      (4,8), (6,8) ]) ] )
gap> u:=u.subgroups;;
```

We now compute the orbits of all of these groups on 3-sets. None fits the pattern expexted.

```
gap> sets:=Combinations([1..8],3);;Length(sets);
56
gap> List(u,h->List(Orbits(h,sets,OnSets),Length));
[ [ 32, 16, 8 ], [ 32, 16, 8 ], [ 32, 24 ], [ 32, 24 ], [ 48, 8 ] ]
```