# Potential Savings in VIFF

Martin Geisler

BRICS
Department of Computer Science
Aarhus University

December 3rd, 2008

# Saving Potentials

- VIFF uses simple pure Python implementations of:
  - Shamir secret sharing
  - Field arithmetic
  - Pseudo-random secret sharing

# Saving Potentials

- VIFF uses simple pure Python implementations of:
  - Shamir secret sharing
  - Field arithmetic
  - Pseudo-random secret sharing

? How much could be saved by reimplementing as a Python extension?

! Difficult to tell without writing the extension. But we can pretend calls to the extension takes no time...

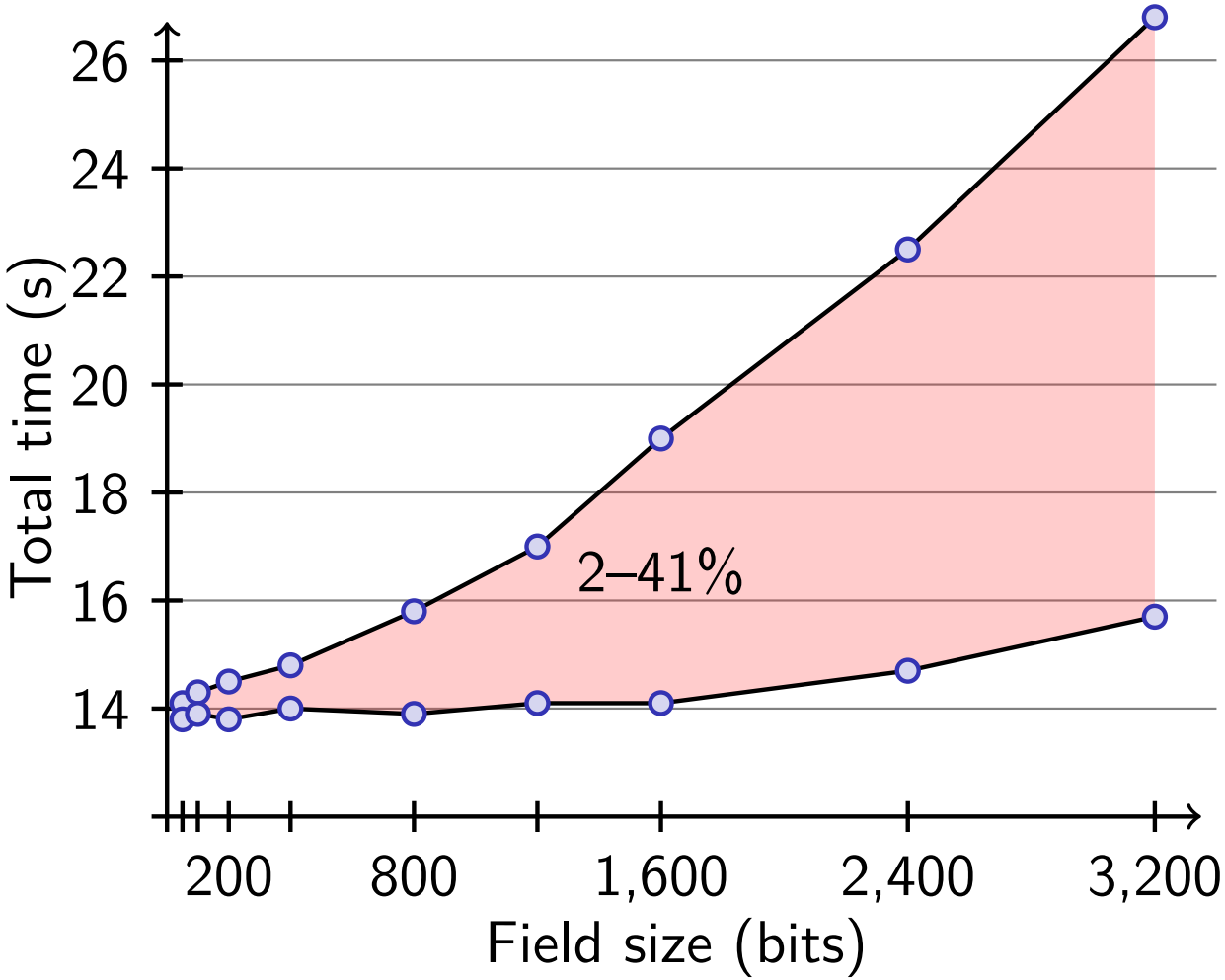
# Fake Field Arithmetic

- Field arithmetic done by `viff.field`
- Creates fake fields using `FakeGF` function:

```
>>> from viff.field import FakeGF
>>> F = FakeGF(1031)
>>> a = F(123)
>>> b = F(234)
>>> a + b
{{1030}}
>>> a * b
{{1030}}
>>> a.sqrt()
{{1030}}
>>> a.bit(100)
1
```

- All operations simply always return $p - 1$ for $\mathbb{Z}_p$.
- Tested using 10,000 multiplications between three machines

# Fake Field Arithmetic – Results



# Fake Shamir Sharing and Recombination

- Shamir secret sharing done by `viff.shamir`
- Replaced by trivial degree-zero sharings:

```
>>> viff.shamir.share(17, 1, 3)
[(1, 17), (2, 17), (3, 17)]
>>> viff.shamir.recombine([(1, 17), (2, 17), (3, 17)])
17
```

- No polynomials, no arithmetic.

# Fake Shamir Sharing and Recombination – Results