When e-th roots become easier than factoring

Antoine Joux[1], David Naccache[2], Emmanuel Thomé[3]

Echternach, January 2008

[1]DGA and UVSQ
[2]ENS
[3]LORIA

Key question

Security of plain RSA

?

Factoring

Quick reminder: RSA

- RSA: Rivest, Shamir, Adleman (1977)
- Public key: N a large integer, e encryption exponent
- Private key: $N = pq$, p and q prime, d decryption exponent

$$ed = \lambda(p-1)(q-1) + 1.$$

$$\text{Encryption} \quad : \quad x \longrightarrow x^e \pmod N$$

$$\text{Decryption} \quad : \quad y \longrightarrow \sqrt[e]{y} \pmod N$$
$$y \longrightarrow y^d \pmod N$$

Quick reminder: RSA and factoring ?

- Pros:
  - Finding d is as difficult as factoring N
    - Probabilistic (already in RSA from Miller 1975)
    - Deterministic (May 2004)
  - Breaking RSA may be as difficult as factoring (Brown 2006)
- Cons:
  - Specific weaknesses:
    - Multiplicative attacks
    - Blinding
  - Breaking RSA may be easier than factoring (Boneh, Venkatesan, 1998)

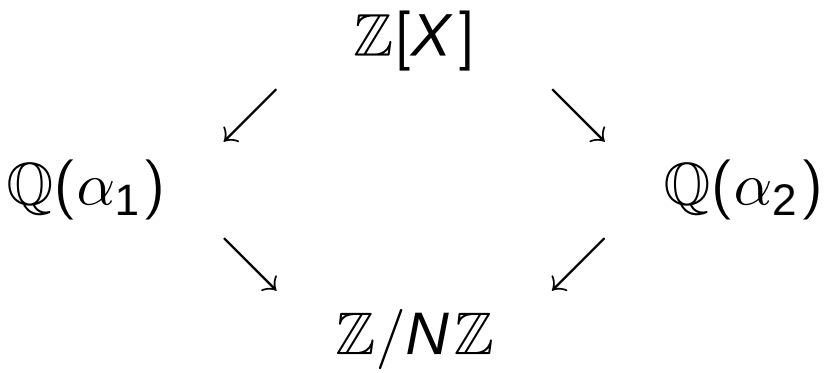
Specific weaknesses

- Multiplicative attacks:
  - From $\sqrt[e]{a}$ and $\sqrt[e]{b}$, deduce $\sqrt[e]{ab}$.
- Blinding:
  - Ask $\sqrt[e]{ar^e}$. Deduce $\sqrt[e]{a}$.

Reformulating the key question

- Given access to an e-th root oracle:
- Can we learn to compute e-th roots ?
  - Efficienty (with a cost lower than factoring) ?

Reminder: Number Field Sieve



- Number fields defined from two polynomials: f_1 and f_2
- Relies on multiplicative relations over smoothness bases
- Applicable to factoring and discrete logarithm
- Complexity:

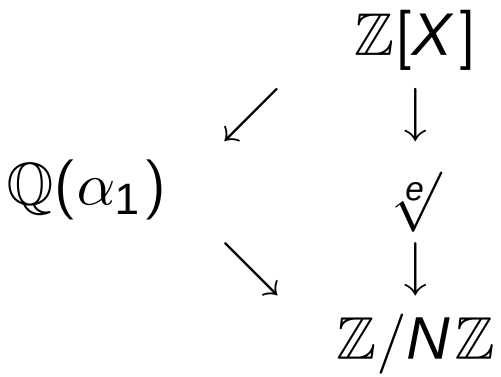
$$L_N(1/3, (64/9)^{1/3}) = e^{((64/9)^{1/3}+o(1))\log^{1/3} N \log\log^{2/3} N)}$$

Reminder: Number Field Sieve

1. Find smooth objects and write multiplicative relations
2. Do linear algebra
3. Final stage
   - Finish factorization: Square root of ideal (Montgomery)
   - Compute individual discrete logarithms: Descent

A special case: Affine modular roots (AMR)

- Special oracle $\sqrt[e]{c+x}$ (c fixed, x small)
  - Multiplicative attack ?
    - Known attacks when $x \geq N^{1/3}$ is allowed
  - Arbitrary e-th roots ?

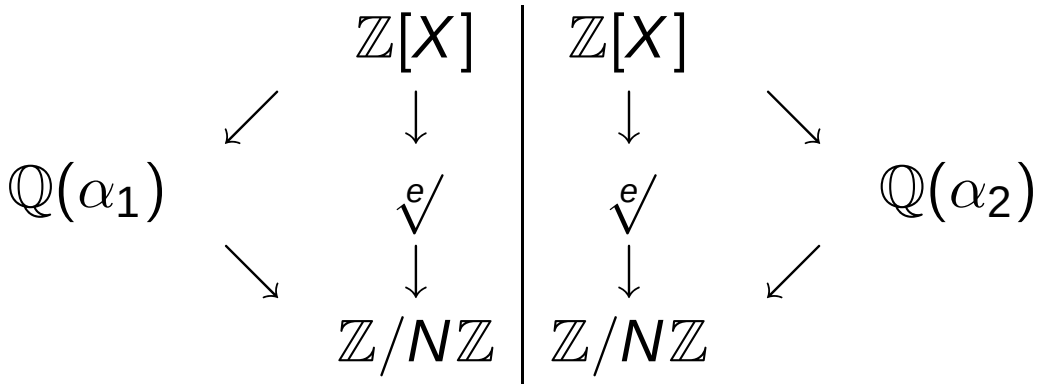


A special case: Affine modular roots

1. One sided smooth objects: multiplicative relations with $\sqrt[e]{\ }$
2. Do linear algebra: $\sqrt[e]{\ }$ of basis elements
3. Final stage
   - Get multiplicative relation
     - Existential forgery
   - Compute arbitrary e-th roots (with additional queries)
     - Universal forgery

Answering the key question

- General oracle $\sqrt[e]{x}$
- Collect two sides
  - Sieving on one side. Twice.
  - Same complexity !



Answering the key question

1. Collect relations:
   - On side 1
   - On side 2: directly obtain e-th roots of basis elements
2. Do linear algebra: $\sqrt[e]{}$ of basis elements (side 1)
3. Optionally enlarge smoothness bases
4. Final stage
   - Descent as in discrete logs
   - Plus e-th roots of ideal (generalized Montgomery)

Complexity analysis

- At best:

$$L(1/3, (32/9)^{1/3}) \approx L(1/3, 1.526)$$

- With basis extension (more practical):

$$L(1/3, 1.577)$$

- Without Montgomery e-th root:

$$L(1/3, (9/2)^{1/3}) \approx L(1/3, 1.651)$$

- Using only an AMR oracle:

$$L(1/3, (6)^{1/3}) \approx L(1/3, 1.817)$$

- Factoring:

$$L(1/3, (64/9)^{1/3}) \approx L(1/3, 1.923)$$

Experiment on 512 bits

With public exponent $e = 65537$.

- Initial sieving: 2 CPU hours[1]
- Linear algebra[2]: 6 hours on 4 proc.
- Bases extension: 44 CPU hours
- Total number of oracle queries: 400 000 000
- Descent time: around one hour
- Montgomery e-th root: five minutes

Reminder: Factoring this number took 8000 mips.years

[1] Timings on AMD Opteron 2.4GHz.

[2] Intel core-2 2.667GHZ

New result (with R. Lercier)

- Technique applicable to discrete logarithm based systems
  - e-th roots problem replaced by static Diffie-Hellman
  - Applicable to all finite fields (improving the $L(1/3)$ constant)
  - No need for Montgomery's algorithm

Open problem

- With this result: e-th roots easier than factoring
- Can we use an e-th root oracle to factor faster ?

Conclusion

Questions ?