Shorter Compact Representations in Real Quadratic Fields

Michael J. Jacobson, Jr.
University of Calgary

Joint work with Alan Silvester and Hugh Williams

West Coast Number Theory 2013

Fundamental Units

$\Delta \in \mathbb{Z}^{>0}$, $\Delta \equiv 0, 1 \pmod 4$ — quadratic discriminant

$\mathbb{Q}(\sqrt{\Delta}) = \{x + y\sqrt{\Delta} \mid x, y \in \mathbb{Q}\}$ — real quadratic field

- h_Δ — ideal class number
- ε_Δ — fundamental unit

Problem: fundamental unit is too large to work with (in general)

- expect $\log \varepsilon_\Delta \approx \sqrt{\Delta}$
- can't write down in time polynomial in $\log \Delta$

Eg. for $\Delta_c = 410286423278424$, coefficients of ε_{Δ_c} have about 103200 decimal digits each.

Compact Representations

Lagarias and Cohen: respresent ε_Δ as a power-product of smaller elements of $\mathbb{Q}(\sqrt{\Delta})$.

Formalized by Buchmann, Thiel, and Williams (1991)

- size polynomial in $\log(\Delta)$

Applications:

- explicit arithmetic with large elements of $\mathbb{Q}(\sqrt{\Delta})$ (norm, multiplication, x or y mod p, ...)
- proof that computing h_Δ is in $NP \cap coNP$ (assuming generalized Riemann hypothesis)

Algorithm

Uses arithmetic with reduced principal ideals (π), $\pi \in \mathbb{Q}(\sqrt{\Delta})$

Idea: "binary exponentiation" using exponent $\log_2 \theta$

- write $\log_2 \theta \approx b_0 2^l + b_1 2^{l-1} + \cdots + b_l$
- given (π_j) with

$$\log_2 \pi_j \approx s_j = \sum_{i=0}^{j} b_i 2^{j-i}$$

compute (π_{j+1}) with

$$\log_2 \pi_{j+1} \approx s_{j+1} = 2s_j + b_{j+1}$$

by reducing $(\pi_j)^2$ and "adjusting"

- have $(\pi_j^2)(\lambda_{j+1}) = (\pi_{j+1})$ for some $\lambda_{j+1} \in \mathbb{Q}(\sqrt{\Delta})$

Compact Representation: Definition

Previous procedure yields compact representation

$$\theta = \pi_k = \prod_{i=0}^{l} \lambda_i^{2^{l-i}}$$

Size of a compact representation:

- number of terms: $O(\log_2 \log_2 \theta)$
- size of each term: $O(\log_2 \Delta)$
- total: $O((\log_2 \log_2 \theta) \log_2 \Delta)$

Eg. compact representation of ε_{Δ_c} has 1212 bits.

Reducing the Sizes of Terms

(π'_{j+1}), the reduction of (π_j^2), has $\log_2 \pi'_{j+1} < 2 \log_2 \pi_j$

- adjustment term λ_{j+1} has to make up for this shortfall, and is larger than desired

Idea: adjust exponentiation so that after reduction, $\log \pi'_{j+1}$ is closer to the target

- aim for $2s_i + b_{i+1} - h$, where reduction shortfall is $\approx h$
- almost all terms have size $O(\log_2 \Delta^{3/4})$
- size of resulting compact representation: $O((\log_2 \log_2 \theta) \log_2 \Delta^{3/4})$

Eg. compact representation of ε_{Δ_c} requires 974 bits

Reducing the Number of Terms

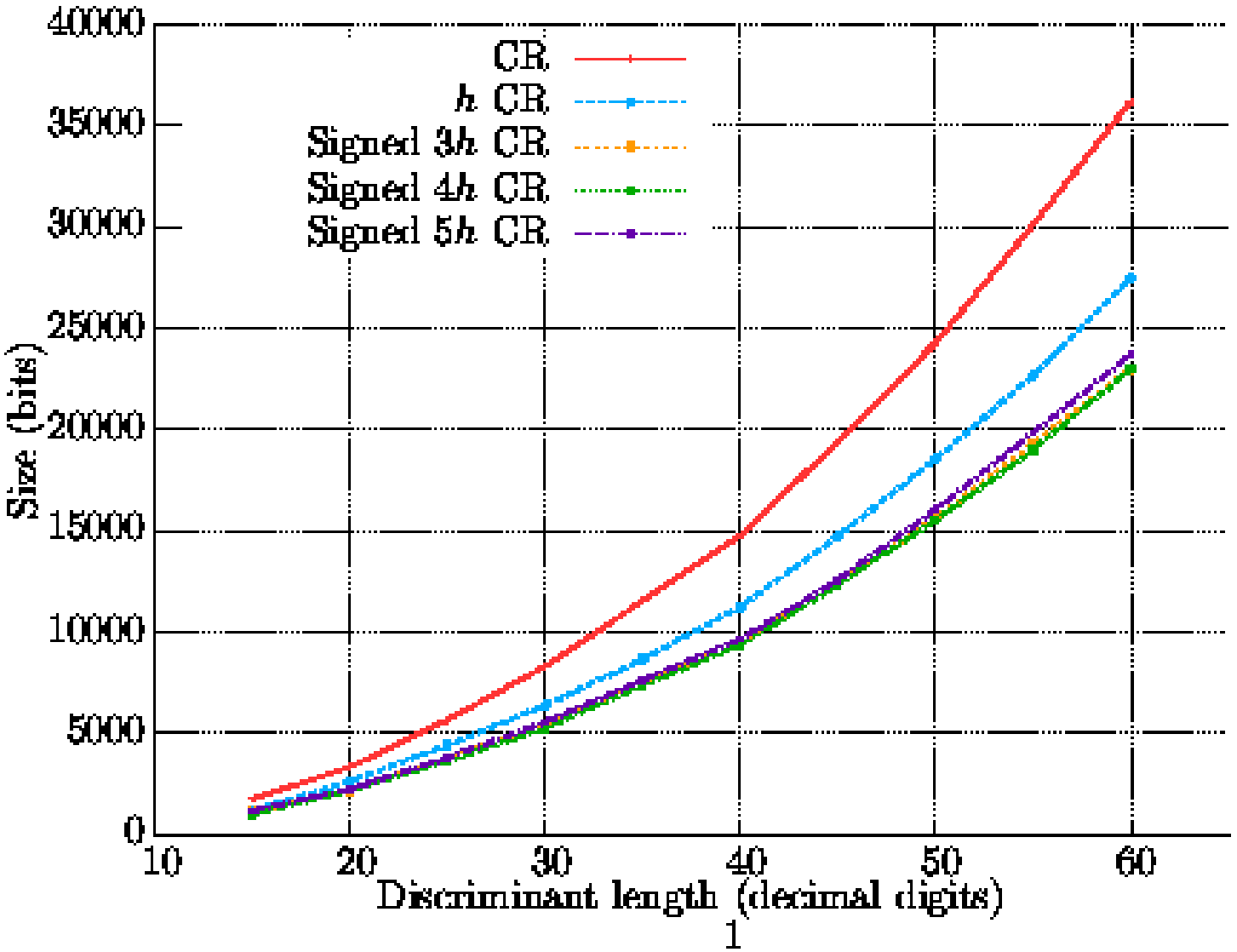
Use signed base x expansion of $\log_2 \theta$

- fewer terms in the power product: $O(\log_x \log_2 \theta)$
- each term is larger (have to reduce xth power): $O(\log_2 \Delta^{\frac{x+1}{4}})$
- size of resulting compact representation: $O((\log_x \log_2 \theta) \log_2 \Delta^{\frac{x+1}{4}})$

Dominant term (as function of x) is $\frac{x+1}{4 \log_2 x}$: min. between $x = 3$ and 4

Eg. using $x = 3$, size of compact representation of ε_{Δ_c} reduces to 843 bits.

Numerical Results



Conclusions

Further improvemens?

- seems hard to reduce size of terms further
- double-base representations to reduce number of terms?
- size sublinear in $\log \Delta$?

Other settings?

- general number fields
- function fields over finite fields