

COMPARATIVE ANALYSIS OF MULTI-PRECISION
ARITHMETIC LIBRARIES FOR PUBLIC KEY
CRYPTOGRAPHY

By

Ashraf Abusharekh
MS CpE Candidate
Department of Electrical & Computer Engineering

Objective

- Evaluation of Multi-precision Arithmetic Libraries for Use in Public Key Cryptography
- Practical Recommendations
 - Which library is best for a particular application?

Outline

- Public Key Cryptosystems
- Libraries
- Evaluation Criteria
- Results
- Conclusions
- Improvements/Future Work

Public Key Cryptosystems

- 2 Different keys, Public Key and Private Key
- Based on trapdoor One-Way Functions
 - Easy to compute
 - Hard to inverse
 - Mathematical problems that guarantee the hardness of inversion:
 - Integer Factorization Problem (IFP)
 - Discrete Logarithm Problem (DLP)
 - Elliptic Curve Discrete Logarithm Problem (ECDLP)
- Public Key Cryptography Security
 - How Hard is the Mathematical Problem?

Public Key Cryptosystems

Basis

| Mathematical Problems | IFP | DLP | ECDLP |
|-------------------------------------|--|---|--|
| Given | $N = p \cdot q$, where p and q are unknown primes | g, p and $y = g^x \bmod p$ where p is a known prime | P , and $Q = kP$ P is a point on Elliptic Curve |
| Find | p and q | x | k |
| PKC based on a given problem | RSA | DH/DSA | ECDH/ECDSA |

Public Key Cryptosystems

Security

- Public Key Related Mathematically to Private Key
- Security Prerequisites
 - Computing Private Key From Public Key Must be infeasible.
- One-Way function difficult to inverse

Public Key Cryptosystems

Best Known Attacks

| PKC | RSA | DSA | ECDSA |
|---|------------------------------------|---|---------------------------|
| One Way Function | $N = p \cdot q$ | $y = g^x \text{ mod } p$ | $Q = kP$ |
| Problem to solve in order to derive the Private Key | IFP | DLP | ECDLP |
| Best Know Attack | General Number Field Sieve GNFS | a. GNFS b. Parallel Collision Search | Parallel Collision Search |
| Size in Bits | N 768 – 2048 | g 768 – 2048 x 160 | k 140 – 224 |

Public Key Schemes

Digital Signature

| Signature Scheme | RSA | DSA | ECDSA |
|-------------------|--|--|--|
| System Parameters | N/A | <p>p, q are primes $2^{159} < q < 2^{160}$ $2^{L-1} < p < 2^L, q p-1$ g generator with order q</p> | E, with order r |
| Key Generation | <p>$2^{L-1} \leq p < 2^L$ $2^{L-1} \leq q < 2^L$ $N = p \cdot q,$ $\phi = (p-1)(q-1)$ e, typically 3 or $(2^{16}+1)$ $\text{GCD}(e, \phi) = 1$ $d = e^{-1} \text{ mod } \phi$ Public (e), Private (d).</p> | <p>random x $0 < x < q$ $y = g^x \text{ mod } p$ Public (y) Private (x).</p> | <p>random s $1 \leq s \leq r - 1$ $W = s \cdot G$ Public (W) Private (s).</p> |

Public Key Schemes

Digital Signature

| Signature Scheme | RSA | DSA | ECDSA |
|------------------------|--|---|---|
| Signature Generation | $s = m^d \bmod N$ Signature (s) | $1 \leq k \leq q - 1$ $r = (g^k \bmod p) \bmod q$ $s = k^{-1} (\text{SHA-1}(m) + x \cdot r) \bmod q$ Signature (r, s) | $1 \leq k \leq r - 1$ $R(x_R, y_R) = k \cdot G$ $c = x_R \bmod r$ $d = k^{-1} \cdot (\text{SHA-1}(m) + x \cdot c) \bmod r$ Signature (c, d) |
| Signature Verification | $m = s^e \bmod N$ | $w = s^{-1} \bmod q$ $u1 = w \cdot \text{SHA-1}(m) \bmod q$ $u2 = r \cdot w \bmod q$ $v = (g^{u1} \cdot y^{u2} \bmod p) \bmod q$ | $w = d^{-1} \bmod r$ $u1 = \text{SHA-1}(m) \cdot w \bmod r$ $u2 = c \cdot w \bmod r$ $V = u1 \cdot G + u2 \cdot W,$ $c' = x_V \bmod r$ |
| Key Size (in bits) | L=768-2048 | L=768-2048 size of $q = 160$ | size of $r = 140-224$ |

Public Key Schemes

Operations

| Signature Scheme | RSA | DSA | ECDSA |
|------------------------|--|---|--|
| System Parameters | N/A | Primality Testing | EC Generation Point Counting |
| Key Generation | <u>Modular Exponentiation</u> Multiplication GCD xGCD | <u>Modular Exponentiation</u> | <u>Scalar Multiplication</u> |
| Signature Generation | <u>Modular Exponentiation</u> | Addition Multiplication <u>Modular Exponentiation</u> xGCD | <u>Scalar Multiplication</u> Addition Multiplication xGCD |
| Signature Verification | <u>Modular Exponentiation</u> | Multiplication <u>Modular Exponentiation</u> xGCD | <u>Scalar Multiplication</u> Multiplication xGCD Point Addition |

Public Key Schemes

Operations

- Large Integers (768 – 2048)
 - Addition, Multiplication, Modular Exponentiation, GCD, xGCD, Primality Testing
- EC Points (140 – 224)
 - EC Point Addition, EC Scalar Multiplication

Problem: Difficult to Implement

Solution: Use Existing Libraries

Libraries

- Many Arithmetic and Number Theoretic Libraries, Commercial and Public Domain
 - Beecrypt, BIGNUM, Botan, bnlib, CLN, cryptolib, CryptoPP, freelib, GMP, Libgcrypt, LiDIA, linteger, MIRACL, nettle, NTL, OpenSSL, PARI, PIOLOGIE, zen
- **Problem:** Which one to use?

Libraries

| | | |
|---|----------|-----|
| Multi-precision, Number theoretic | CLN | C++ |
| | GMP | C |
| | LiDIA | C++ |
| | NTL | C++ |
| | PIOLOGIE | C++ |
| Cryptographic Primitives and Schemes | CryptoPP | C++ |
| | MIRACL | C |
| | OpenSSL | C |

Evaluation Criteria

- Cost
- Documentation & Ease of Use
- Supported Compilers
- Support for PKC
 - Primitive Operations
 - PK Schemes
- Performance of Primitive Operations

Evaluation Criteria

Cost

| | Free for Commercial Use |
|-----------------|--------------------------------|
| CLN | YES |
| CryptoPP | YES |
| GMP | YES |
| LiDIA | NO |
| MIRACL | NO |
| NTL | YES |
| OpenSSL | YES |
| PIOLOGIE | NO |

Evaluation Criteria

Documentation & Ease of Use

| | | | | |
|----------------------|---------------------|--------------|-------------------------------|-------------|
| Documentation | Sufficient | | GMP LiDIA MIRACL NTL | Best |
| | | | CLN | PIOLOGIE |
| | Insufficient | Worst | CryptoPP | OpenSSL |
| | | Hard | Ease of use | Easy |

Evaluation Criteria

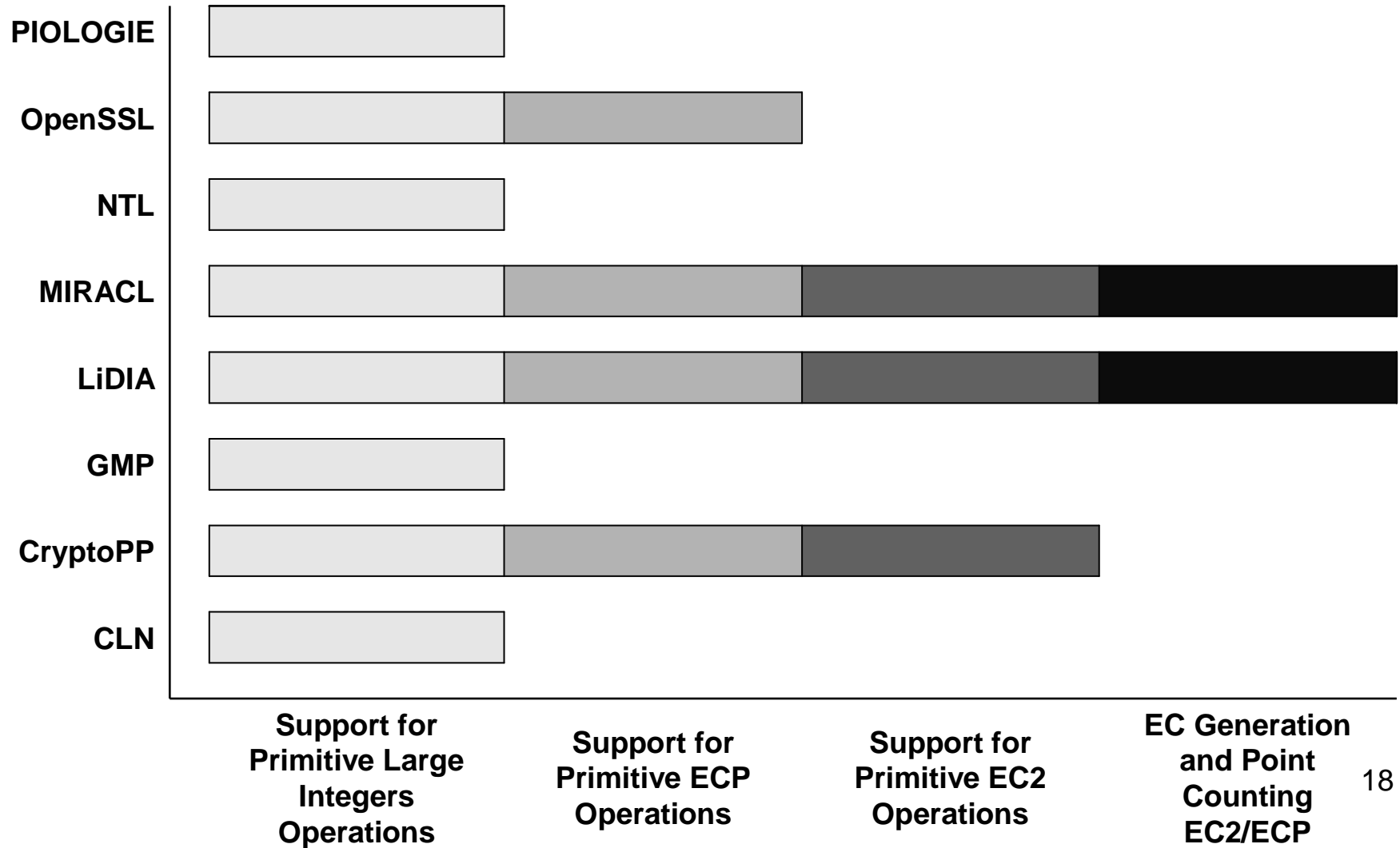
Supported Compilers

| | | | | | | | |
|------------------|-------------------------|-------------------------------|---------------|-----------|-----------|-----------|-----------|
| SUN WorkShop C++ | | | | | | | |
| MIPSpro C++ | | | | | | | |
| MSVC | | | | | | | |
| KAI C++ | | | | | | | |
| VisualAge C++ | | | | | | | |
| IBM CSet++ | | | | | | | |
| IBM C++ | | | | | | | |
| HP C++ | | | | | | | |
| HP aC++ | | | | | | | |
| GNU C/C++ | MSVC | MSVC | | | | | |
| Digital C++ | Intel C/C++ | Microsoft eMbedded Visual C++ | MSVC | | | | |
| front end | GNU C/C++ | GNU C/C++ | Intel C/C++ | | | | |
| Borland C/C++ | Sun WorkShop, Forte C++ | DEC C | GNU C/C++ | | | | |
| Apogee C++ | CodeWarrior Pro | CodeWarrior Pro | Borland C/C++ | MSVC | | | |
| Watcom C++ | Borland C++ Builder | Borland C/C++ | ARM C | GNU C/C++ | GNU C/C++ | GNU C/C++ | GNU C/C++ |
| PIOLOGIE | CryptoPP | OpenSSL | MIRACL | NTL | CLN | GMP | LiDIA |

Evaluation Criteria

Support for PKC

➤ Primitive Operations



Evaluation Criteria

Support for PKC Schemes

| |
|-----|
| RSA |
| DSA |
| DH |

OpenSSL

| |
|----------|
| RW |
| RSA |
| PV |
| NR2/PV |
| NR |
| MQV |
| ECPV |
| ECNR2/PV |
| ECNR |
| ECMQV |
| ECDSA |
| ECDH |
| DSA |
| DH |

MIRACL

| |
|---------|
| XTR-DH |
| RW |
| RSA |
| Rabin |
| NR |
| MQV |
| LUC |
| ESIGN |
| ElGamal |
| ECNR |
| ECMQV |
| ECIES |
| ECDSA |
| ECDH |
| DSA |
| DLIES |
| DH |

CryptoPP 19

Evaluation Criteria

Performance of Primitive Operations

Operations

- Large Integers (768/1024/2048)
 - Multiplication, Modular Exponentiation, GCD, xGCD, Primality Testing
- ECP (162/224/384)
 - EC Point Addition, EC Scalar Multiplication
- EC2 (163/233/409)
 - Addition, Scalar Multiplication

Evaluation Criteria

Performance of Primitive Operations

Platforms

| Processor/hardware | Operating System | Compiler |
|---|----------------------|---------------------|
| 2.00 GHz Pentium IV Processor, 512 MB RAM | Windows XP Cygwin | GNU C/C++ 3.3.1 |
| | RedHat Linux 9.0 | GNU C/C++ 3.3.1 |
| Sun: 2x 400 MHz UltraSPARC-Solaris-II, 4-MB E-cache, 2048 MB RAM | Solaris 5.8 | GNU C/C++ 2.95.2 |

Evaluation Criteria

Performance of Primitive Operations

Methodology

- Measuring Performance of Operations
 - RDTSC Method: Clock Cycles
 - Pentium IV: RDTSC Instruction
 - Timing Method: Milliseconds
 - UltraSPARC-II, NO RDTSC instruction,
“gettimeofday()”
 - 100 execution times for each operation

Evaluation Criteria

Performance of Primitive Operations

Methodology

➤ Inputs

- Large Integers:
 - Random $I_n, J_n, K_n, n = \{768, 1024, 2048\}$
 - Random Primes $P_n^{(j)}, n = \{768, 1024, 2048\}, j = [0, 9]$
- EC2
 - SEC 2 recommended 163, 233, 409.
 - Random Points $T_n, S_n, n = \{163, 233, 409\}$
- ECP
 - Random, 162, 226, 386.
 - Random Points $T_n, S_n, n = \{162, 226, 386\}$

Evaluation Criteria

Performance of Primitive Operations

Methodology

➤ Raw Results

| Operation | | |
|---------------------------------|---|--|
| Multiplication | $\text{result}_n = I_n * J_n$ | Minimum over 100 execution times |
| Mod Exp E=3 | $\text{result}_n = (I_n)^3 \bmod K_n$ | |
| Mod Exp E=65537 | $\text{result}_n = (I_n)^{65537} \bmod K_n$ | |
| Mod Exp Large E | $\text{result}_n = (I_n)^{J_n} \bmod K_n$ | |
| EC Point Addition | $\text{result}_n = T_n + S_n$ | |
| EC Scalar Multiplication | $\text{result}_n = (r - 2) T_n$ | |
| Primality | $\text{result}_n = \text{IsPrime}(P_n^{(j)})$ | 1- Minimum: over 100 execution times for each $P_n^{(j)}$ 2- Average: over 10 $P_n^{(j)}$ |
| GCD | $\text{result}_n = \text{GCD}(P_n^{(j)}, K_i)$ | |
| xGCD | $\text{result}_n = \text{xGCD}(P_n^{(j)}, K_i)$ | |

Evaluation Criteria

Performance of Primitive Operations

Methodology

➤ Operation Ranking

| P4-WinXP/MULTIPLICATION(Clock Cycles) | | | |
|---------------------------------------|------------------------------|-------------------------------|-------------------------------|
| Library | <i>result</i> ₇₆₈ | <i>result</i> ₁₀₂₄ | <i>result</i> ₂₀₄₈ |
| CLN | 8,940 | 11,763 | 29,133 |
| CryptoPP | 38,432 | 37,928 | 78,755 |
| GMP | 3,423 | 5,364 | 17,605 |
| LiDIA | 3,573 | 6,047 | 18,722 |
| MIRACL | 10,974 | 18,613 | 71,512 |
| NTL | 3,381 | 5,426 | 17,722 |
| OpenSSL | 9,055 | 15,218 | 48,438 |
| Piologie | 29,910 | 41,163 | 113,977 |
| <i>Min</i> _n | 3,381 | 5,364 | 17,605 |

| P4-WinXP/MULTIPLICATION Ranking | | | | |
|---------------------------------|-------------------------|--------------------------|--------------------------|------|
| Library | <i>R</i> ₇₆₈ | <i>R</i> ₁₀₂₄ | <i>R</i> ₂₀₄₈ | Rank |
| CLN | 2.64 | 2.19 | 1.65 | 2.12 |
| CryptoPP | 11.37 | 7.07 | 4.47 | 7.11 |
| GMP | 1.01 | 1.00 | 1.00 | 1.00 |
| LiDIA | 1.06 | 1.13 | 1.06 | 1.08 |
| MIRACL | 3.25 | 3.47 | 4.06 | 3.58 |
| NTL | 1.00 | 1.01 | 1.01 | 1.01 |
| OpenSSL | 2.68 | 2.84 | 2.75 | 2.75 |
| Piologie | 8.85 | 7.67 | 6.47 | 7.60 |

$$\text{Operation Rank/Lib/OS} = \sqrt[3]{\prod_{n=768,1024,2048} R_n} \quad \text{where } R_n = \frac{\text{result}_n}{\text{Min}_n}$$

Evaluation Criteria

Performance of Primitive Operations

Methodology

➤ Library Ranking

| | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 | |
|----------|-------|-------|-----------|---------|--------|--------|---------------|
| Library | MUL | E = 3 | E = 65537 | Large E | GCD | xGCD | P4-WinXP Rank |
| CLN | 2.12 | 2.23 | 2.25 | 2.79 | 1.34 | 1.37 | 1.95 |
| CryptoPP | 7.11 | 15.17 | 4.71 | 4.04 | 464.90 | 9.99 | 14.56 |
| GMP | 1.00 | 1.00 | 1.00 | 1.00 | 1.01 | 1.08 | 1.01 |
| LiDIA | 1.08 | 1.45 | 1.08 | 1.65 | 1.03 | 1.10 | 1.21 |
| MIRACL | 3.58 | 22.40 | 4.56 | 2.62 | 5.15 | 3.15 | 5.00 |
| NTL | 1.01 | 1.42 | 1.17 | 1.18 | 1.00 | 1.00 | 1.12 |
| OpenSSL | 2.75 | 8.07 | 2.65 | 2.33 | 8.31 | 12.17 | 4.90 |
| PIOLOGIE | 7.60 | 7.40 | 6.63 | 10.65 | 16.41 | 213.30 | 15.51 |

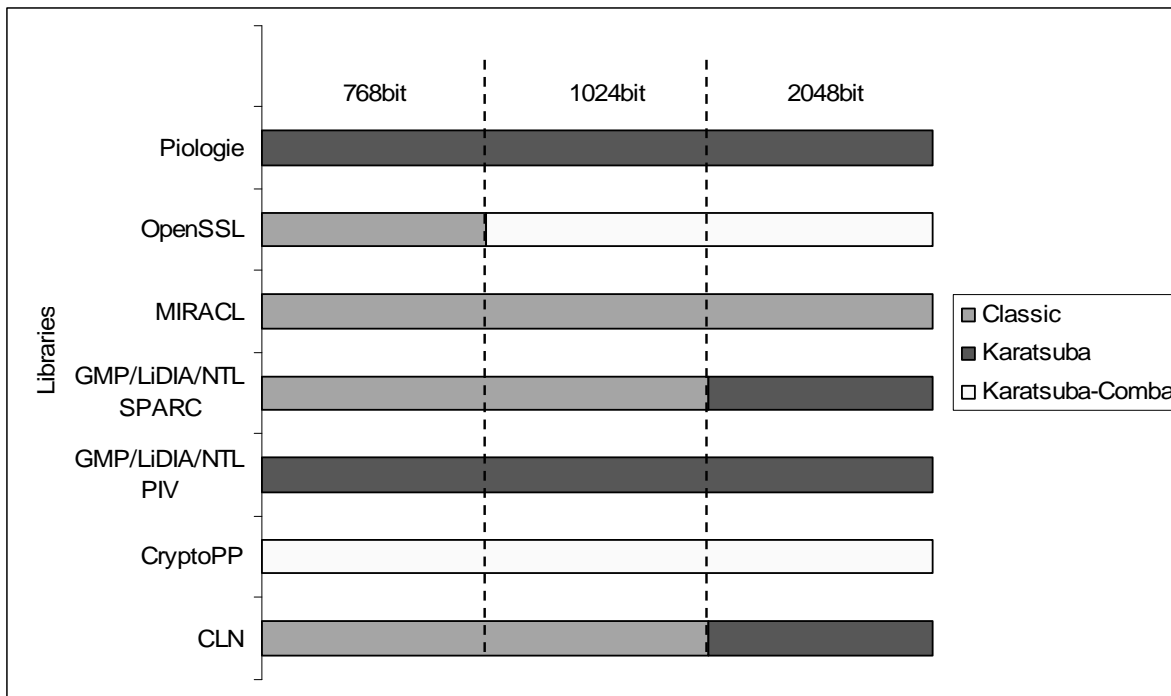
$$\text{Library Rank/OS} = \sqrt[N]{\prod_{k=1}^N X_k}$$

Factors Affecting Performance

- Low level routines
 - Targeting Pentium 4 and UltraSPARC
- Algorithms
 - Choice of Algorithm
 - Different Implementations

Factors Affecting Performance

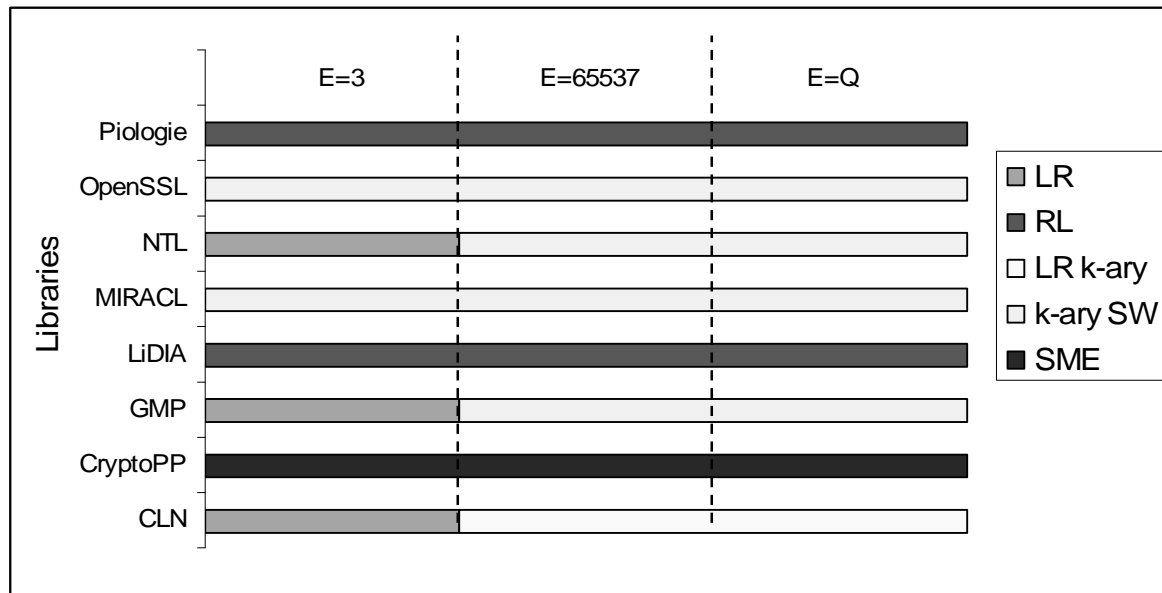
Examples



| P4-WinXP/MULTIPLICATION (Clock Cycles) | | | |
|---|------------------------------|-------------------------------|-------------------------------|
| Library | <i>result</i> ₇₆₈ | <i>result</i> ₁₀₂₄ | <i>result</i> ₂₀₄₈ |
| CLN | 8,940 | 11,763 | 29,133 |
| CryptoPP | 38,432 | 37,928 | 78,755 |
| GMP | 3,423 | 5,364 | 17,605 |
| LiDIA | 3,573 | 6,047 | 18,722 |
| MIRACL | 10,974 | 18,613 | 71,512 |
| NTL | 3,381 | 5,426 | 17,722 |
| OpenSSL | 9,055 | 15,218 | 48,438 |
| Piologie | 29,910 | 41,163 | 113,977 |
| <i>Min</i>_n | 3,381 | 5,364 | 17,605 |

Factors Affecting Performance

Examples



| Library | P4-WinXP/P ³ MOD N Clock Cycles | | |
|------------------------|---|-------------------------------|-------------------------------|
| | <i>result</i> ₇₆₈ | <i>result</i> ₁₀₂₄ | <i>result</i> ₂₀₄₈ |
| CLN | 60,221 | 76,916 | 174,066 |
| CryptoPP | 513,080 | 511,291 | 972,970 |
| GMP | 21,162 | 33,346 | 103,581 |
| LiDIA | 37,316 | 49,645 | 120,347 |
| MIRACL | 426,733 | 716,643 | 2,685,718 |
| NTL | 42,500 | 43,462 | 113,399 |
| OpenSSL | 178,888 | 265,801 | 807,107 |
| Piologie | 185,394 | 243,061 | 658,572 |
| <i>Min_n</i> | 21,162 | 33,346 | 103,581 |

LR: Left-to-Right

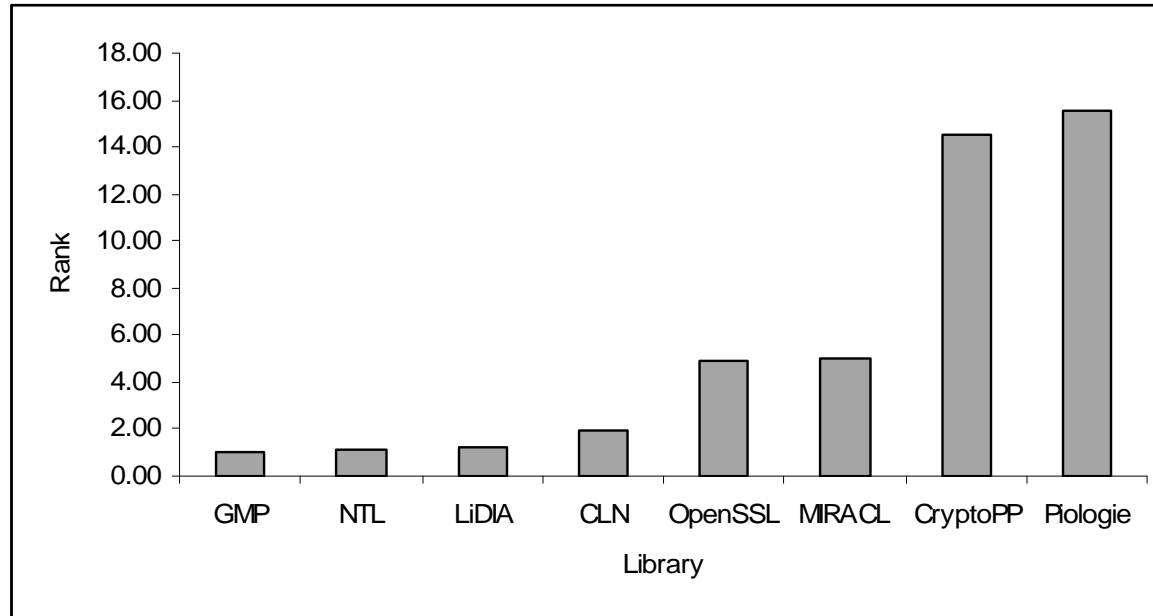
RL: Right-to-Left

SME: Simultaneous Multiple Exponentiation

SW: Sliding Window

Performance Results - Integers

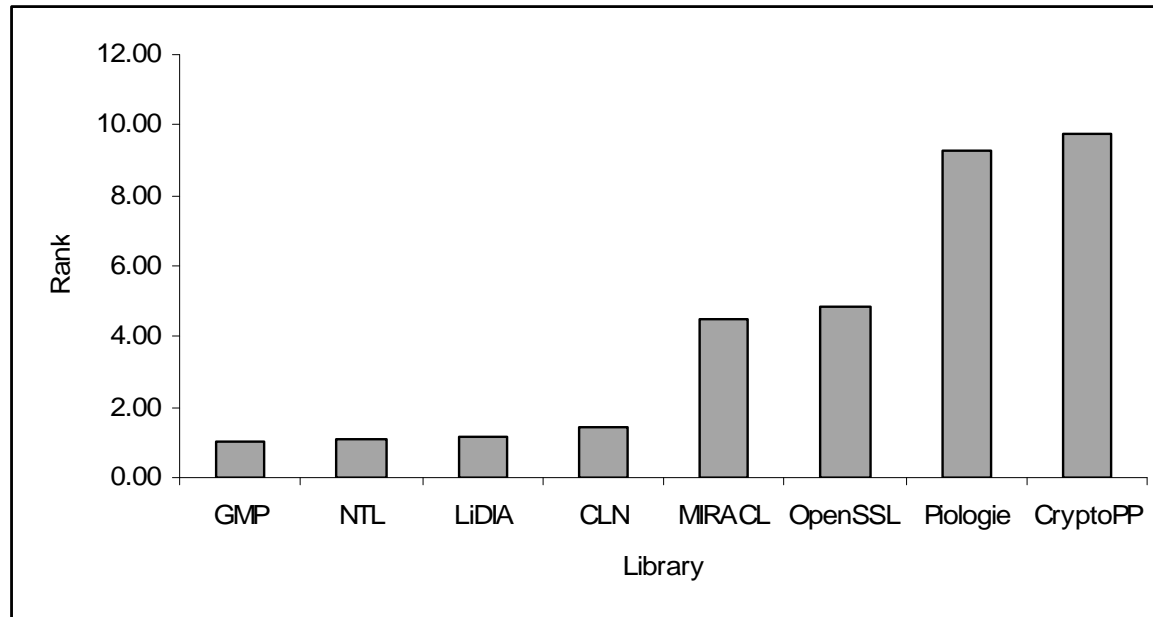
P4-WinXP



| Library | MUL | E = 3 | E = 65537 | Large E | GCD | xGCD | P4-WinXP Rank |
|----------|------|-------|-----------|---------|--------|--------|---------------|
| CLN | 2.12 | 2.23 | 2.25 | 2.79 | 1.34 | 1.37 | 1.95 |
| CryptoPP | 7.11 | 15.17 | 4.71 | 4.04 | 464.90 | 9.99 | 14.56 |
| GMP | 1.00 | 1.00 | 1.00 | 1.00 | 1.01 | 1.08 | 1.01 |
| LiDIA | 1.08 | 1.45 | 1.08 | 1.65 | 1.03 | 1.10 | 1.21 |
| MIRACL | 3.58 | 22.40 | 4.56 | 2.62 | 5.15 | 3.15 | 5.00 |
| NTL | 1.01 | 1.42 | 1.17 | 1.18 | 1.00 | 1.00 | 1.12 |
| OpenSSL | 2.75 | 8.07 | 2.65 | 2.33 | 8.31 | 12.17 | 4.90 |
| PIOLOGIE | 7.60 | 7.40 | 6.63 | 10.65 | 16.41 | 213.30 | 15.51 |

Performance Results - Integers

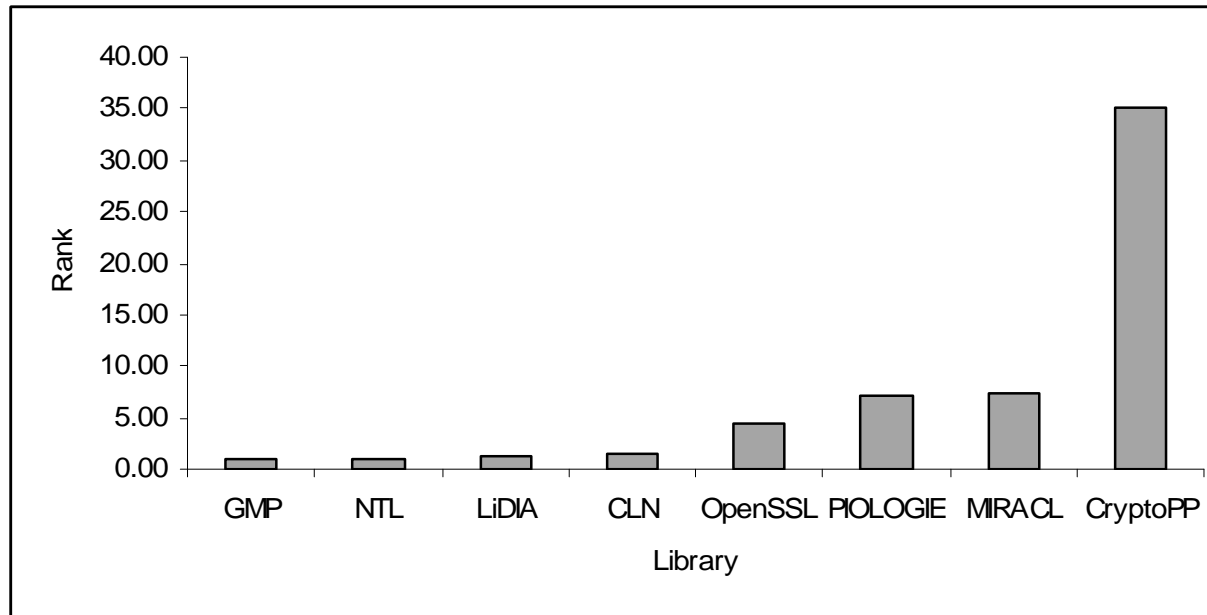
P4-RedHat



| Library | MUL | E = 3 | E = 65537 | Large E | GCD | xGCD | P4-RH Rank |
|----------|------|-------|-----------|---------|-------|-------|------------|
| CLN | 1.50 | 1.27 | 1.39 | 1.87 | 1.37 | 1.27 | 1.43 |
| CryptoPP | 4.49 | 9.19 | 3.79 | 5.04 | 65.96 | 16.82 | 9.78 |
| GMP | 1.00 | 1.00 | 1.01 | 1.00 | 1.00 | 1.08 | 1.01 |
| LiDIA | 1.00 | 1.10 | 1.06 | 1.84 | 1.00 | 1.09 | 1.15 |
| MIRACL | 3.60 | 21.06 | 4.30 | 2.77 | 3.99 | 2.36 | 4.52 |
| NTL | 1.01 | 1.20 | 1.10 | 1.29 | 1.01 | 1.00 | 1.10 |
| OpenSSL | 2.80 | 7.12 | 2.43 | 2.43 | 8.93 | 12.49 | 4.86 |
| PIOLOGIE | 5.35 | 5.01 | 5.07 | 8.79 | 21.95 | 24.22 | 9.27 |

Performance Results - Integers

UltraSPARC-Solaris



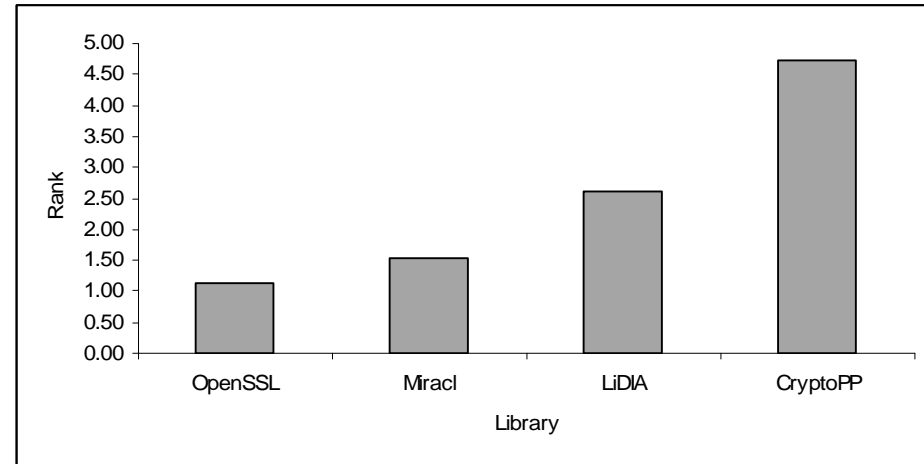
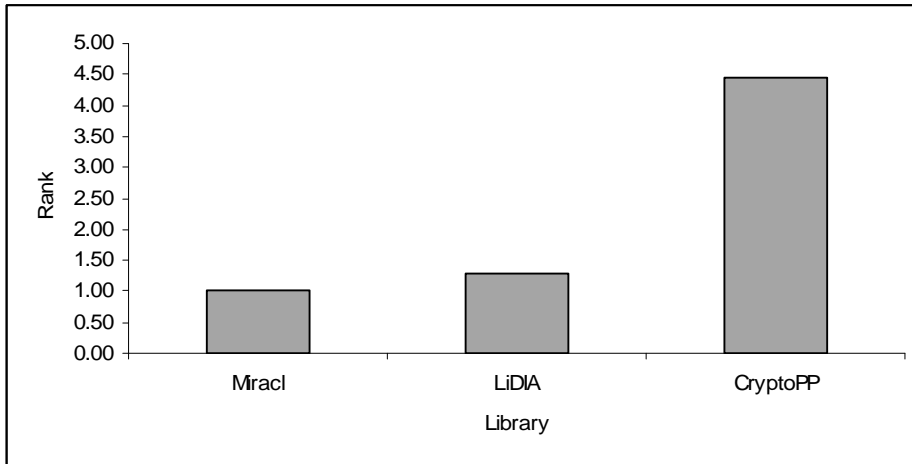
| Library | MUL | E = 3 | E = 65537 | Large E | GCD | xGCD | SPARC Rank |
|----------|-------|-------|-----------|---------|--------|-------|------------|
| CLN | 1.21 | 1.60 | 1.70 | 1.98 | 1.67 | 1.40 | 1.58 |
| CryptoPP | 16.43 | 38.52 | 18.10 | 17.68 | 184.68 | 49.08 | 34.99 |
| GMP | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.12 | 1.02 |
| LiDIA | 1.00 | 1.20 | 1.10 | 1.55 | 1.02 | 1.14 | 1.16 |
| MIRACL | 9.08 | 23.85 | 2.98 | 7.41 | 8.77 | 3.71 | 7.33 |
| NTL | 1.00 | 1.25 | 1.15 | 1.13 | 1.05 | 1.00 | 1.09 |
| OpenSSL | 2.16 | 7.80 | 2.81 | 2.45 | 7.67 | 7.74 | 4.36 |
| PIOLOGIE | 3.51 | 4.13 | 4.06 | 5.95 | 9.13 | 37.22 | 7.01 |

Performance Results - EC

P4-WinXP

EC2

ECP



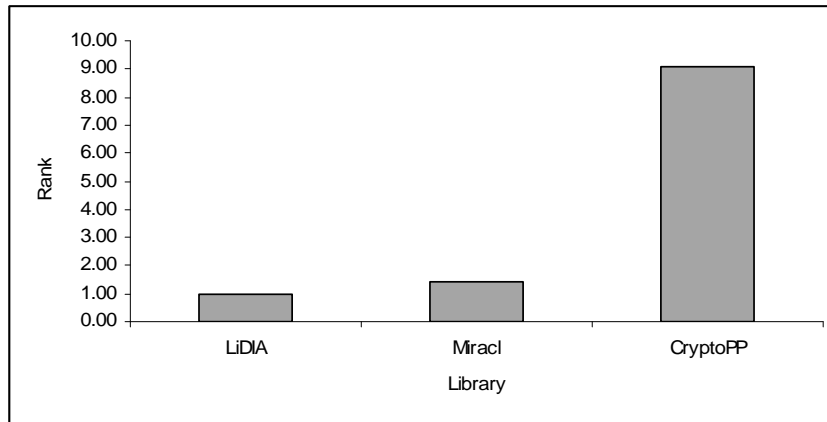
| Library | ADD | Scalar MUL | P4-WinXP Rank |
|----------|------|------------|---------------|
| CryptoPP | 4.56 | 4.35 | 4.46 |
| LiDIA | 1.33 | 1.22 | 1.28 |
| MIRACL | 1.00 | 1.00 | 1.00 |

| Library | ADD | Scalar MUL | P4-WinXP Rank |
|----------|------|------------|---------------|
| CryptoPP | 6.02 | 3.72 | 4.73 |
| LiDIA | 1.97 | 3.45 | 2.61 |
| MIRACL | 1.05 | 2.24 | 1.53 |
| OpenSSL | 1.30 | 1.00 | 1.14 |

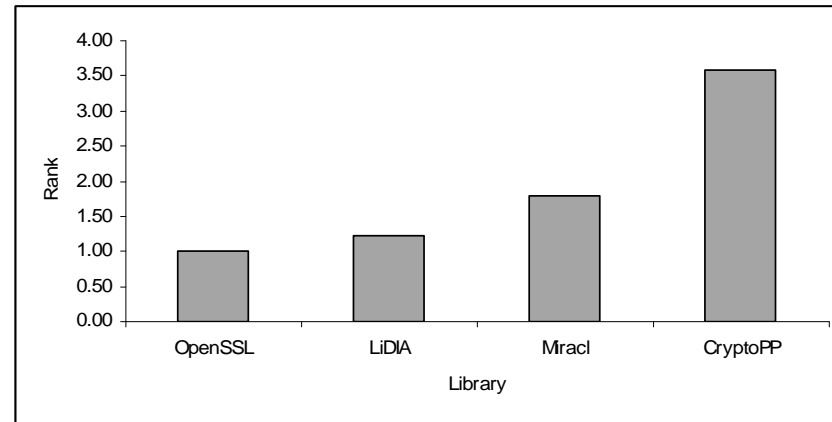
Performance Results - EC

P4-RedHat

EC2



ECP



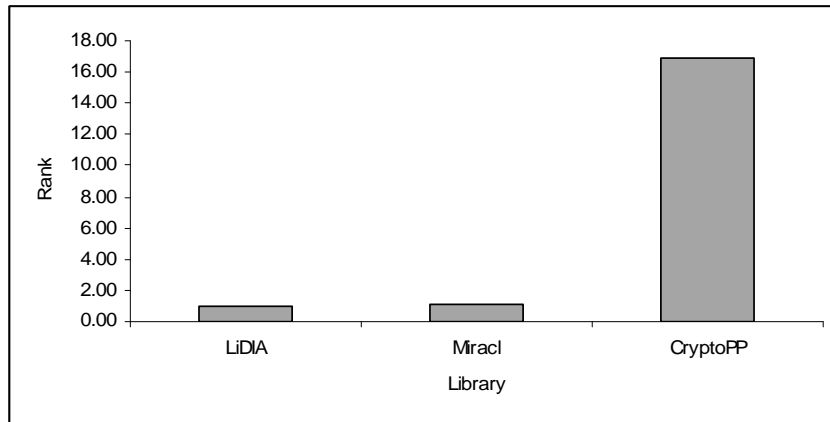
| Library | ADD | Scalar MUL | P4-RedHat Rank |
|----------|------|------------|----------------|
| CryptoPP | 9.55 | 8.67 | 9.10 |
| LiDIA | 1.00 | 1.00 | 1.00 |
| MIRACL | 1.48 | 1.32 | 1.40 |

| Library | ADD | Scalar MUL | P4-RedHat Rank |
|----------|------|------------|----------------|
| CryptoPP | 5.40 | 2.38 | 3.58 |
| LiDIA | 1.06 | 1.44 | 1.24 |
| MIRACL | 1.61 | 2.02 | 1.80 |
| OpenSSL | 1.02 | 1.00 | 1.01 |

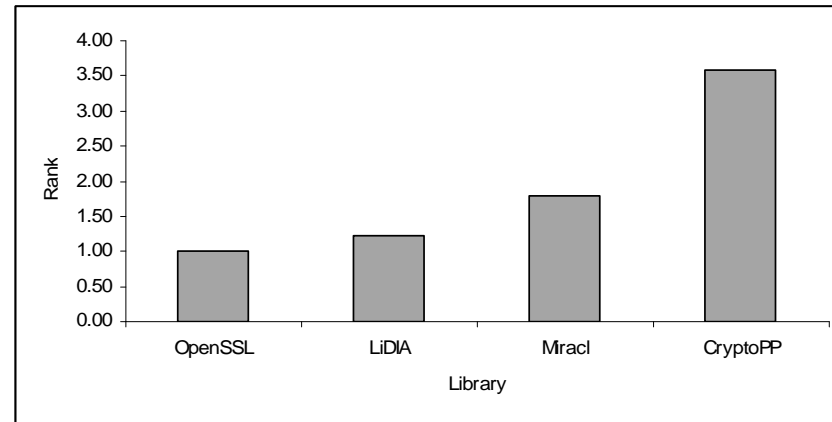
Performance Results - EC

UltraSPARC-Solaris

EC2



ECP



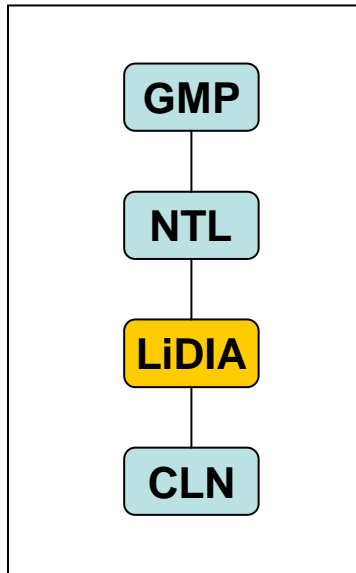
| Library | ADD | Scalar MUL | UltraSPARC-Solaris Rank |
|----------|-------|------------|-------------------------|
| CryptoPP | 17.10 | 16.65 | 16.87 |
| LiDIA | 1.00 | 1.00 | 1.00 |
| MIRACL | 1.17 | 1.15 | 1.16 |

| Library | ADD | Scalar MUL | UltraSPARC-Solaris Rank |
|----------|------|------------|-------------------------|
| CryptoPP | 9.46 | 7.15 | 8.23 |
| LiDIA | 1.05 | 1.73 | 1.35 |
| MIRACL | 1.49 | 2.60 | 1.97 |
| OpenSSL | 1.00 | 1.00 | 1.00 |

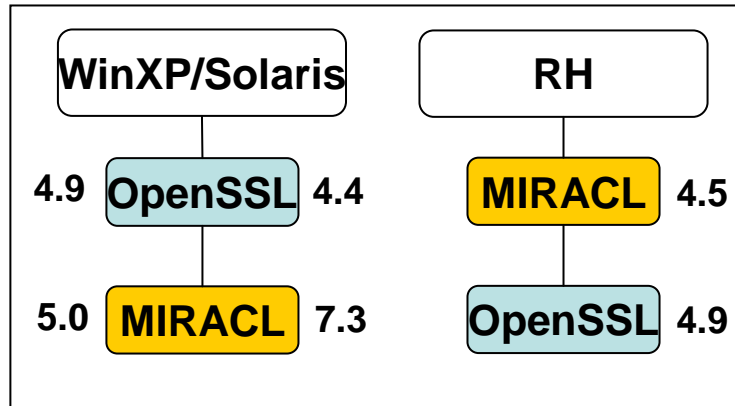
Results

Operations On Large Integers

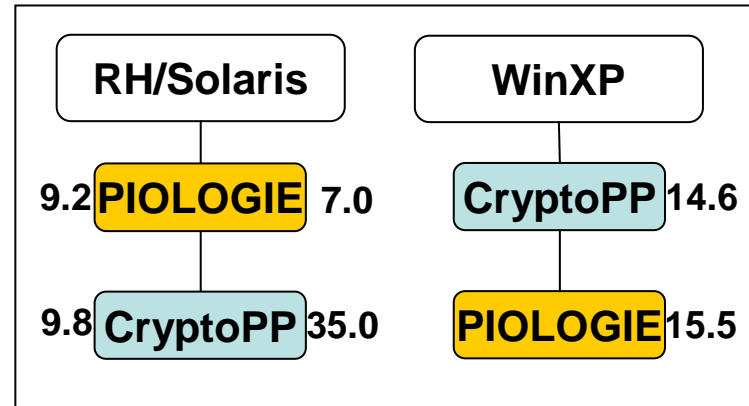
Fast



Medium

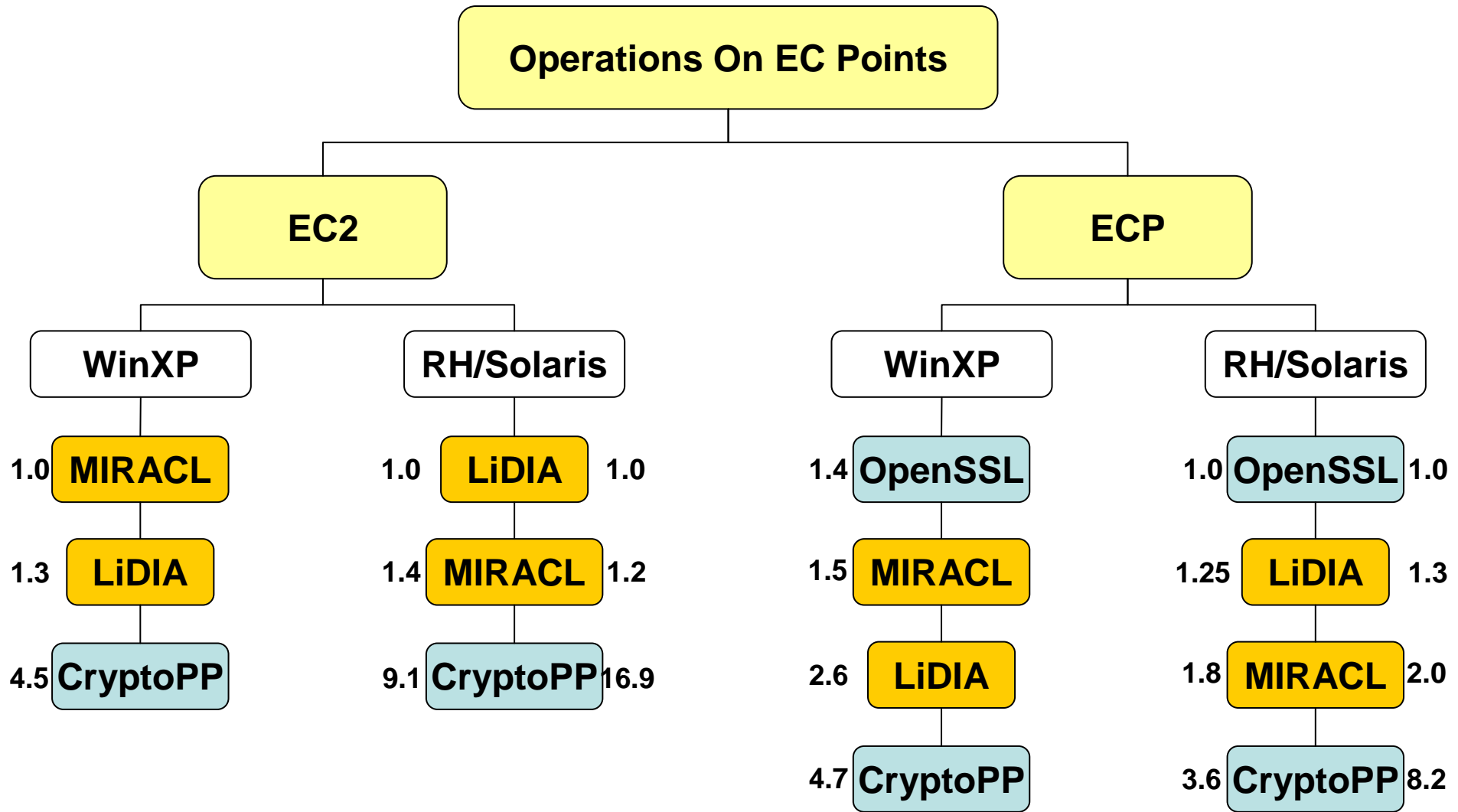


Slow



Free

Results



Free

Conclusions

Which library is best for a particular application?

- Best Performance
- Highest Support => less Time spent for development
- Lowest Cost

No one Library Offers All three.

Which library is best for implementing PK Schemes operating on
LARGE INTEGERS?

| | | | |
|-------------|------|-------------------------------|--------------------------|
| Performance | High | GMP, NTL, <u>LiDIA</u> CLN | Best |
| | | | OpenSSL <u>MIRACL</u> |
| | Low | <u>PIOLOGIE</u> | Worst |
| | | Low | High |
| | | Primitives | Schemes |
| | | Support | |

Which library is best for implementing PK Schemes operating on
LARGE INTEGERS?

- High Performance :
 - A lot of time devoted for development (low support)
 - Free
 - Use GMP.
- Medium Performance/Medium Time Devoted For Development :
 - Free
 - Use OpenSSL.
- High Support/Not Enough Time to Develop
 - Free
 - Use CryptoPP.

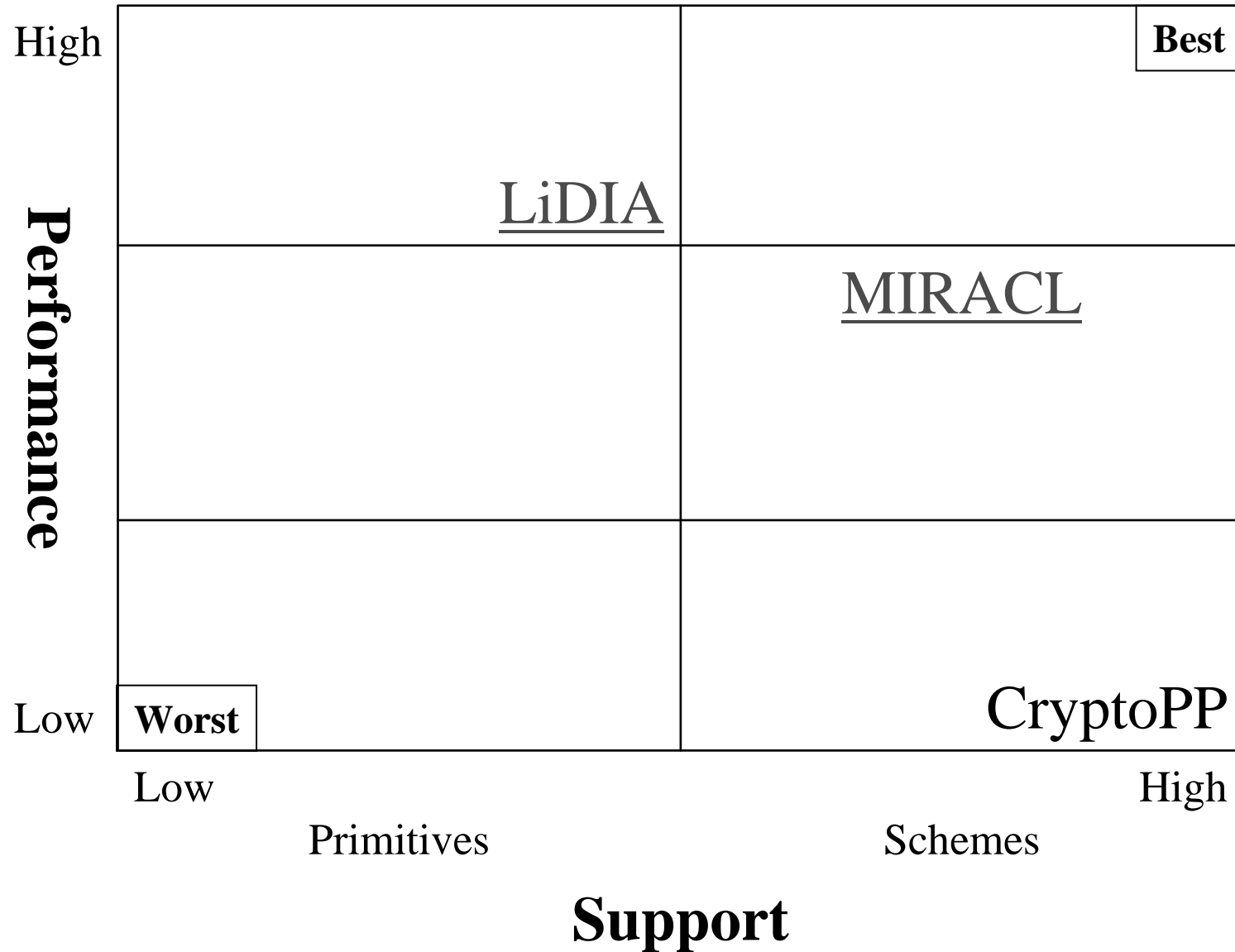
Which library is best for implementing PK Schemes operating on ECP?

| | | | | |
|--------------------|------|----------------|--------------|---------------|
| Performance | High | OpenSSL | | Best |
| | | | <u>LiDIA</u> | |
| | | | | <u>MIRACL</u> |
| Low | | Worst | | CryptoPP |
| | | Low | | High |
| | | Primitives | | Schemes |
| | | Support | | |

Which library is best for implementing PK Schemes operating on ECP?

- High Performance :
 - A lot of time devoted for development (low support)
 - Free
 - Use OpenSSL.
- Medium Performance/Good Support
 - Not Free
 - Use MIRACL.
- High Support/Not Enough Time to Develop
 - Free
 - Use CryptoPP.

Which library is best for implementing PK Schemes
operating on EC2?



Which library is best for implementing PK Schemes operating on EC2?

- High Performance :
 - A lot of time devoted for development (low support)
 - Not Free
 - Use LiDIA.
- Medium Performance/Good Support
 - Not Free
 - Use MIRACL.
- High Support/Not Enough Time to Develop
 - Free
 - Use CryptoPP.

Improvements/Future Work

- Implement PK Schemes
 - Ranking based on PK Schemes
- More Libraries, Platforms.

Contributions

- Cryptography and Network-Security Implementation lab
 - Development of KRYPTOS, Educational Software used in two Cryptography Courses ECE 646 and ECE 746

Thank You

Questions