Renaissance of Precomputation in a Post-Quantum World

Aydin Aysu, Patrick Schaumont

Virginia Tech

aydinay@vt.edu

04/03/15
Introduction

The changes in a PQ-world:
- Cryptanalysis tools
- Security primitives
- Embedded systems

Precomputation as an optimization methodology
- Previous ([Koyama92],[Brickell92],[Rooij95])
- Recent ([Bernstein12],[Ateniese13],[Bianchi14])
- Apply it on post-quantum digital signatures
- Quantify its effect on energy, latency and system yield
Introduction

The changes in a PQ-world:
- Cryptanalysis tools
- Security primitives
- Embedded systems

Precomputation as an optimization methodology
- Previous ([Koyama92],[Brickell92],[Rooij95])
- Recent ([Bernstein12],[Ateniese13],[Bianchi14])
- Apply it on post-quantum digital signatures
- Quantify its effect on energy, latency and system yield
The changes in a PQ-world:
- Cryptanalysis tools
- Security primitives
- Embedded systems

Precomputation as an optimization methodology
- Previous ([Koyama92],[Brickell92],[Rooij95])
- Recent ([Bernstein12],[Ateniese13],[Bianchi14])

Apply it on post-quantum digital signatures
Quantify its effect on energy, latency and system yield
The changes in a PQ-world:
- Cryptanalysis tools
- Security primitives
- Embedded systems

Precomputation as an optimization methodology
- Previous ([Koyama92],[Brickell92],[Rooij95])
- Recent ([Bernstein12],[Ateniese13],[Bianchi14])
- Apply it on post-quantum digital signatures
- Quantify its effect on energy, latency and system yield
Introduction

Motivation

Renaissance of Precomputation

Precomputation requires extra preparatory operations and extra storage

The case for precomputation

Memory: 15 new generations of flash memory in 20 years = 25000× cost improvement [Harari11]

Energy: Harvesting platforms towards a greener future

Energy profile (extrapolated from [Bianchi’13])

Improves latency, run-time energy, availability and yield
Precomputation requires extra preparatory operations and extra storage.

The case for precomputation:

- Memory: 15 new generations of flash memory in 20 years
  \[= 25000 \times \text{cost improvement} \ [\text{Harari11}]\]

- Energy: Harvesting platforms towards a greener future

Energy profile (extrapolated from [Bianchi’13])

Improves latency, run-time energy, availability and yield.
Precomputation requires extra preparatory operations and extra storage

The case for precomputation

Memory: 15 new generations of flash memory in 20 years
$= 25000 \times \text{cost improvement}$ [Harari11]

Energy: Harvesting platforms towards a greener future

Energy profile (extrapolated from [Bianchi’13])

Improves latency, run-time energy, availability and yield
Precomputation requires extra preparatory operations and extra storage

The case for precomputation

- **Memory:** 15 new generations of flash memory in 20 years
  \[= 25000 \times \text{cost improvement} \quad \text{[Harari11]}\]
- **Energy:** Harvesting platforms towards a greener future

Energy profile (extrapolated from [Bianchi’13])

Improves latency, run-time energy, availability and yield
Defining the Execution Modes

Separate operations into two phases: \textit{offline} and \textit{online}.

Precompute during the offline phase.

Minimize the length (latency) of the online phase.
Defining the Execution Modes

Separate operations into two phases: offline and online.

Precompute during the offline phase.

Minimize the length (latency) of the online phase.
Defining the Execution Modes

Separate operations into two phases: offline and online

Precompute during the offline phase

Minimize the length (latency) of the online phase
Winternitz Hash-based Signatures

Precompute intermediate nodes
Start from the closest node
Winternitz Hash-based Signatures

Precompute intermediate nodes
Start from the closest node
Precomputing PQ-signatures

GLP Lattice-based Signatures

Precompute nonce coupons
Spend during the online phase
GLP Lattice-based Signatures

Precompute nonce coupons
Spend during the online phase
GLP Lattice-based Signatures

Precompute nonce coupons
Spend during the online phase
Energy harvesting setup with precise energy and execution time measurements
Results

Energy Profiling

GLP requires less energy than Winternitz

ω = 8 requires less energy than ω = 4
GLP requires less energy than Winternitz

$\omega = 8$ requires less energy than $\omega = 4$
GLP has lower latency than Winternitz

$\omega = 4$ has lower latency than $\omega = 8$
GLP has lower latency than Winternitz

\[ \omega = 4 \text{ has lower latency than } \omega = 8 \]
Winternitz signature yield

Significant improvement for critical energy levels

$3 \times$ more signatures for full battery
Winternitz signature yield

Significant improvement for critical energy levels
3 × more signatures for full battery
GLP signature yield

Significant improvements for critical energy levels

1.5× more signatures for full battery
GLP signature yield

Significant improvements for critical energy levels
1.5× more signatures for full battery
Conclusions

Optimizations bring complex algorithms into life on constrained platforms

Precomputation is useful
  Improvements of up to 82x latency, 11x run-time energy and 3x system yield

Precomputation is NOT infeasible
  At least on moderate research platforms

Precomputation is an orthogonal methodology
  Combine with arithmetic and programming optimizations

Real-time embedded systems favor precomputable signatures
  An implementation insight on signatures
Conclusions

Optimizations bring complex algorithms into life on constrained platforms

Precomputation is useful
   Improvements of up to 82x latency, 11x run-time energy and 3x system yield

Precomputation is NOT infeasible
   At least on moderate research platforms

Precomputation is an orthogonal methodology
   Combine with arithmetic and programming optimizations

Real-time embedded systems favor precomputable signatures
   An implementation insight on signatures
Conclusions

Optimizations bring complex algorithms into life on constrained platforms

Precomputation is useful
   Improvements of up to 82x latency, 11x run-time energy and 3x system yield

Precomputation is NOT infeasible
   At least on moderate research platforms

Precomputation is an orthogonal methodology
   Combine with arithmetic and programming optimizations

Real-time embedded systems favor precomputable signatures
   An implementation insight on signatures
Conclusions

Optimizations bring complex algorithms into life on constrained platforms

Precomputation is useful
   Improvements of up to 82x latency, 11x run-time energy and 3x system yield

Precomputation is NOT infeasible
   At least on moderate research platforms

Precomputation is an orthogonal methodology
   Combine with arithmetic and programming optimizations

Real-time embedded systems favor precomputable signatures
   An implementation insight on signatures
Conclusions

Optimizations bring complex algorithms into life on constrained platforms

Precomputation is useful
  Improvements of up to 82x latency, 11x run-time energy and 3x system yield

Precomputation is NOT infeasible
  At least on moderate research platforms

Precomputation is an orthogonal methodology
  Combine with arithmetic and programming optimizations

Real-time embedded systems favor precomputable signatures
  An implementation insight on signatures
Acknowledgements

NSF award no 1314598
Bilgiday Yuce

For more information:
http://eprint.iacr.org/2015/288
Back-up Slides: Application Context

### Computing Device
- **Operation**: Signature generation
- **Platform**: Simple microcontrollers
- **Rate**: 1 signing per hour
- **Optimization**: Latency

### Edge of the Cloud
- **Operation**: Signature verification
- **Platform**: High-end CPUs
- **Rate**: 1000 ver. per minute
- **Optimization**: Throughput
1: procedure KEY GENERATION\((a, s_1, s_2, t)\)
2: \(s_1, s_2 \leftarrow \text{rand}(R_{1}^{p^n})\)
3: \(a \leftarrow \text{rand}(R_{p^n})\)
4: \(t \leftarrow as_1 + s_2\)
5: end procedure
6: procedure SIGNING\((s_1, s_2, \mu, z_1, z_2, c)\)
7: \(y_1, y_2 \leftarrow \text{rand}(R_{k}^{p^n})\)
8: \(c \leftarrow H(ay_1 + y_2, \mu)\)
9: \(z_1 \leftarrow s_1c + y_1, z_2 \leftarrow s_2c + y_2\)
10: if \(z_1\) or \(z_2\) \(\notin R_{k-32}^{p^n}\) go to step 7
11: end procedure
12: procedure VERIFICATION\((z_1, z_2, c, \mu, t, )\)
13: Validate iff
14: \(z_1, z_2 \in R_{k-32}^{p^n}\)
15: \(c = H(az_1 + z_2 + tc, \mu)\)
16: end procedure