NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Improving Speed of Dilithium's Signing Procedure

Prasanna Ravi
G1802146B

School of Computer Science and Engineering
Physical Analysis and Cryptographic Engineering, Temasek Laboratories

17th April 2019

Table of Contents

1. Context
2. Background
3. Algorithmic Optimizations
4. Experimental Results
5. Future Work
6. Conclusion

Table of Contents

1 Context

2 Background

3 Algorithmic Optimizations

4 Experimental Results

5 Future Work

6 Conclusion

Context

- Huge money in quantum computing is being invested by computer industry giants like Google, IBM, Intel and other companies like D-Wave, IonQ.

Context

- Huge money in quantum computing is being invested by computer industry giants like Google, IBM, Intel and other companies like D-Wave, IonQ.
- A large scale quantum computer has the potential to break all of public key cryptography that we use today.

Context

- Huge money in quantum computing is being invested by computer industry giants like Google, IBM, Intel and other companies like D-Wave, IonQ.
- A large scale quantum computer has the potential to break all of public key cryptography that we use today.
- This has prompted the cryptographic community to develop quantum resistant alternatives for public-key cryptography.

Context

- Huge money in quantum computing is being invested by computer industry giants like Google, IBM, Intel and other companies like D-Wave, IonQ.
- A large scale quantum computer has the potential to break all of public key cryptography that we use today.
- This has prompted the cryptographic community to develop quantum resistant alternatives for public-key cryptography.
- NIST process for standardization of Post-Quantum cryptography is underway.

Context

- Huge money in quantum computing is being invested by computer industry giants like Google, IBM, Intel and other companies like D-Wave, IonQ.
- A large scale quantum computer has the potential to break all of public key cryptography that we use today.
- This has prompted the cryptographic community to develop quantum resistant alternatives for public-key cryptography.
- NIST process for standardization of Post-Quantum cryptography is underway.
- Lattice-based cryptography has contributed the maximum number of proposals in terms of post-quantum key exchange and post-quantum signature schemes.

This Work

- `Dilithium` is one of the candidate signature schemes based on lattice-based cryptography.

This Work

- `Dilithium` is one of the candidate signature schemes based on lattice-based cryptography.
- This work involves improving the signing speed of `Dilithium` signature scheme.

This Work

- `Dilithium` is one of the candidate signature schemes based on lattice-based cryptography.
- This work involves improving the signing speed of `Dilithium` signature scheme.
- The Signing procedure is iterative in nature with multiple *rejection* conditions in each iteration.

This Work

- `Dilithium` is one of the candidate signature schemes based on lattice-based cryptography.
- This work involves improving the signing speed of `Dilithium` signature scheme.
- The Signing procedure is iterative in nature with multiple *rejection* conditions in each iteration.
- Several iterations of the signing procedure are repeated until the outputs satisfy a certain condition.
- Repetition rate hampers the performance of the signing procedure.

This Work

- `Dilithium` is one of the candidate signature schemes based on lattice-based cryptography.
- This work involves improving the signing speed of `Dilithium` signature scheme.
- The Signing procedure is iterative in nature with multiple *rejection* conditions in each iteration.
- Several iterations of the signing procedure are repeated until the outputs satisfy a certain condition.
- Repetition rate hampers the performance of the signing procedure.
- We attempt to improve the signing speed through algorithmic optimizations.

Table of Contents

1 Context

2 Background

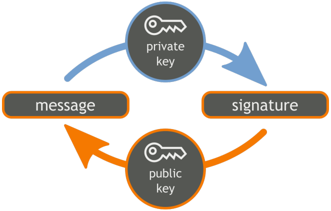
3 Algorithmic Optimizations

4 Experimental Results

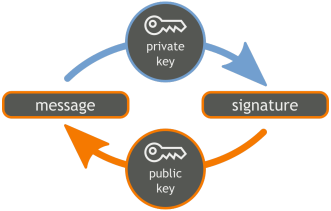
5 Future Work

6 Conclusion

Digital Signature

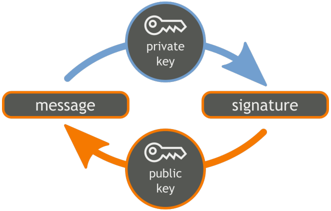


Digital Signature



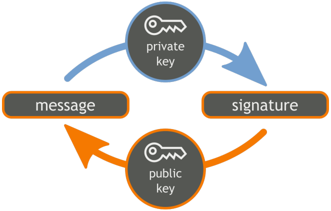
- A signature scheme consists of three procedures:

Digital Signature



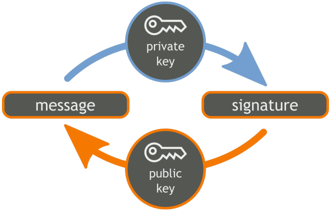
- A signature scheme consists of three procedures:
  - **Key Generation** (Generates the public and private keys)

Digital Signature



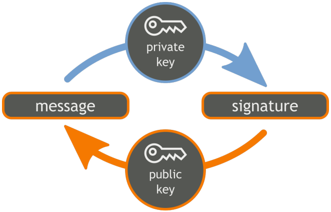
- A signature scheme consists of three procedures:
    - **Key Generation** (Generates the public and private keys)
    - **Signature Generation** (Generates signature for a given message)

Digital Signature



- A signature scheme consists of three procedures:
  - **Key Generation** (Generates the public and private keys)
  - **Signature Generation** (Generates signature for a given message)
  - **Verification** (Verifies correctness of signature)

Digital Signature



- A signature scheme consists of three procedures:
  - **Key Generation** (Generates the public and private keys)
  - **Signature Generation** (Generates signature for a given message)
  - **Verification** (Verifies correctness of signature)

Learning With Errors (LWE) Problem

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$
- Search LWE: Given several pairs $(\mathbf{A}, \mathbf{T})$, find $\mathbf{S}$.

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$
- Search LWE: Given several pairs $(\mathbf{A}, \mathbf{T})$, find $\mathbf{S}$.
- Decisional LWE: Distinguish between valid LWE pairs $(\mathbf{A}, \mathbf{T})$ from uniformly random samples in $(\mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n)$.

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$
- Search LWE: Given several pairs $(\mathbf{A}, \mathbf{T})$, find $\mathbf{S}$.
- Decisional LWE: Distinguish between valid LWE pairs $(\mathbf{A}, \mathbf{T})$ from uniformly random samples in $(\mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n)$.
- Computations over matrices and Vectors were mapped to polynomials in the more efficient variants of LWE such as Ring-LWE (RLWE) and Module-LWE (MLWE).

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$
- Search LWE: Given several pairs $(\mathbf{A}, \mathbf{T})$, find $\mathbf{S}$.
- Decisional LWE: Distinguish between valid LWE pairs $(\mathbf{A}, \mathbf{T})$ from uniformly random samples in $(\mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n)$.
- Computations over matrices and Vectors were mapped to polynomials in the more efficient variants of LWE such as Ring-LWE (RLWE) and Module-LWE (MLWE).
- Ring LWE: $\mathbf{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ with $\mathbf{A}, \mathbf{S}, \mathbf{E} \in \mathbf{R}_q$.

Learning With Errors (LWE) Problem

- Let $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ and $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^n \leftarrow D_\sigma$
- $\mathbf{T} = (\mathbf{A} \times \mathbf{S} + \mathbf{E}) \in \mathbb{Z}_q^n$
- Search LWE: Given several pairs $(\mathbf{A}, \mathbf{T})$, find $\mathbf{S}$.
- Decisional LWE: Distinguish between valid LWE pairs $(\mathbf{A}, \mathbf{T})$ from uniformly random samples in $(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n)$.
- Computations over matrices and Vectors were mapped to polynomials in the more efficient variants of LWE such as Ring-LWE (RLWE) and Module-LWE (MLWE).
- Ring LWE: $\mathbf{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ with $\mathbf{A}, \mathbf{S}, \mathbf{E} \in \mathbf{R}_q$.
- Module LWE: $\mathbf{R}_q^{k\times l} = (\mathbb{Z}_q[X]/(X^n + 1))^{k\times l}$ with $\mathbf{A} \in \mathbf{R}_q^{k\times \ell}$, $\mathbf{S} \in \mathbf{R}_q^{\ell}$, $\mathbf{E} \in \mathbf{R}_q^{k}$.

`Dilithium` Signature Scheme

- Security of `Dilithium` is based on the MLWE problem.
- Computations are performed over *matrices* and *vectors* of polynomials.
- Signature generation is an iterative procedure with multiple rejection conditions.
- Two algorithmic level optimizations to improve signing speed have been explored.
  - *Opt-1:* Reduction of computations in every rejected iteration.
  - *Opt-2:* Reduction of repetition rate.

Table of Contents

1. Context
2. Background
3. Algorithmic Optimizations
4. Experimental Results
5. Future Work
6. Conclusion

Reducing Computations in Rejected Iterations

- The signing procedure consists of a number of conditional checks.

Reducing Computations in Rejected Iterations

- The signing procedure consists of a number of conditional checks.
- Is it possible to detect the rejections early to reduce the overhead of the rejected iterations?

Reducing Computations in Rejected Iterations

- The signing procedure consists of a number of conditional checks.
- Is it possible to detect the rejections early to reduce the overhead of the rejected iterations?
- We perform an *early-evaluation* of the rejection conditions, so we detect the rejections early and immediately abort the current iteration.

Dilithium's Signing Procedure

Procedure Sign (sk, M)
  $\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$
  $\mu = \mathsf{CRH}(\mathsf{tr} \| M)$
  $\kappa = 0, (\mathbf{z}, \mathbf{h}) = \perp$
  while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
    $\mathbf{y} \in S_{\gamma_1 - 1}^{\ell} := \mathsf{ExpandMask}(K \| \mu \| \kappa)$
    $\mathbf{w} = \mathbf{A} \cdot \mathbf{y}$
    $\mathbf{w}_1 = \mathsf{HB}_q(\mathbf{w}, 2\gamma_2)$
    $\mathbf{c} \in B_{60} = H(\mu \| \mathbf{w}_1)$
    $\mathbf{z} = \mathbf{y} + \mathbf{c} \cdot \mathbf{s}_1$
    $(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{D}_q(\mathbf{w} - \mathbf{c} \cdot \mathbf{s}_2, 2\gamma_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}_1 \neq \mathbf{w}_1$ **then**
      $(\mathbf{z}, \mathbf{h}) = \perp$
    else
      $\mathbf{h} = \mathsf{MH}_q(-\mathbf{c} \cdot \mathbf{t}_0, \mathbf{w} - \mathbf{c} \cdot \mathbf{s}_2 + \mathbf{c} \cdot \mathbf{t}_0, 2\gamma_2)$
      if $\|\mathbf{c} \cdot \mathbf{t}_0\|_\infty \geq \gamma_2$ or $\mathsf{wt}(\mathbf{h}) > \omega$ **then**
        $(\mathbf{z}, \mathbf{h}) = \perp$
    end
    $\kappa = \kappa + 1$
  end
  return $\sigma = (\mathbf{z}, \mathbf{h}, \mathbf{c})$

Dilithium's Signing Procedure

Procedure Sign (sk, M)

- $\mathbf{A} \in R_q^{k\times\ell} :=$ ExpandA(ρ)
- $\mu =$ CRH$(\mathrm{tr}\|M)$
- $\kappa = 0, (\mathbf{z}, \mathbf{h}) = \perp$
- **while** $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
  - $\mathbf{y} \in S_{\gamma_1-1}^{\ell} :=$ ExpandMask$(K\|\mu\|\kappa)$
  - $\mathbf{w} = \mathbf{A}\cdot\mathbf{y}$
  - $\mathbf{w}_1 = \mathrm{HB}_q(\mathbf{w}, 2\gamma_2)$
  - $\mathbf{c} \in B_{60} = H(\mu\|\mathbf{w}_1)$
  - $\mathbf{z} = \mathbf{y} + \mathbf{c}\cdot\mathbf{s}_1$
  - $(\mathbf{r}_1, \mathbf{r}_0) := \mathrm{D}_q(\mathbf{w} - \mathbf{c}\cdot\mathbf{s}_2, 2\gamma_2)$
  - **if** $\boxed{\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta}$ or $\boxed{\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta}$ or $\mathbf{r}_1 \neq \mathbf{w}_1$ **then**
    - $(\mathbf{z}, \mathbf{h}) = \perp$
  - **else**
    - $\mathbf{h} = \mathrm{MH}_q(-\mathbf{c}\cdot\mathbf{t}_0, \mathbf{w} - \mathbf{c}\cdot\mathbf{s}_2 + \mathbf{c}\cdot\mathbf{t}_0, 2\gamma_2)$
    - **if** $\|\mathbf{c}\cdot\mathbf{t}_0\|_\infty \geq \gamma_2$ or $\mathrm{wt}(\mathbf{h}) > \omega$ **then**
      - $(\mathbf{z}, \mathbf{h}) = \perp$
  - **end**
  - $\kappa = \kappa + 1$
- **end**
- **return** $\sigma = (\mathbf{z}, \mathbf{h}, \mathbf{c})$

Reducing Computations in Rejected Iterations

- We target the rejection conditions that yield frequent rejections.

Reducing Computations in Rejected Iterations

- We target the rejection conditions that yield frequent rejections.
- Both these rejection conditions are only infinity norm checks ($\|\cdot\|_\infty < K$).

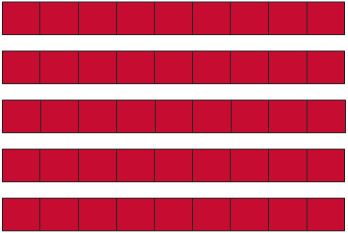
Reducing Computations in Rejected Iterations

- We target the rejection conditions that yield frequent rejections.
- Both these rejection conditions are only infinity norm checks ($\|\cdot\|_\infty < K$).
- The condition has to be satisfied for all coefficients of a given module.

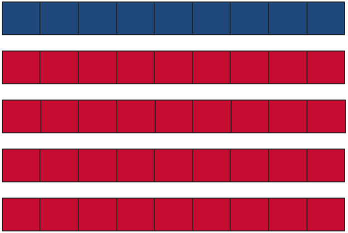
Reducing Computations in Rejected Iterations

- We target the rejection conditions that yield frequent rejections.
- Both these rejection conditions are only infinity norm checks ($\|\cdot\|_\infty < K$).
- The condition has to be satisfied for all coefficients of a given module.
- Consider the computations involving module $\mathbf{z} \in R_q^\ell$.

Evaluation of Rejection Conditions

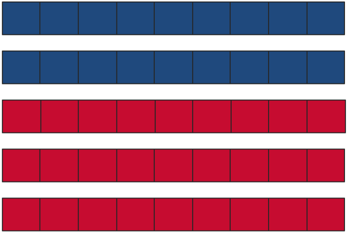


Evaluation of Rejection Conditions



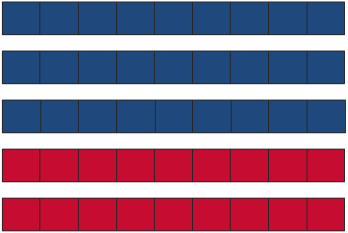
No of Computations: 1

Evaluation of Rejection Conditions



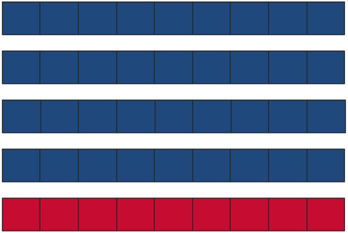
No of Computations: 2

Evaluation of Rejection Conditions



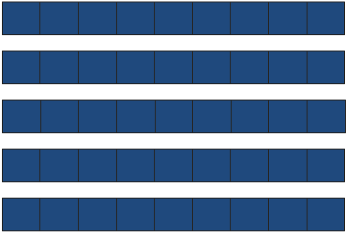
No of Computations: 3

Evaluation of Rejection Conditions



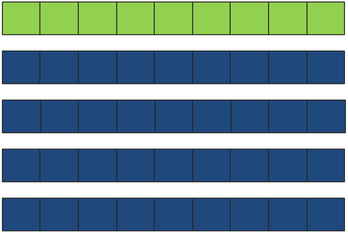
No of Computations: 4

Evaluation of Rejection Conditions



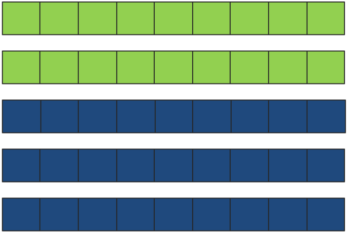
No of Computations: N

Evaluation of Rejection Conditions



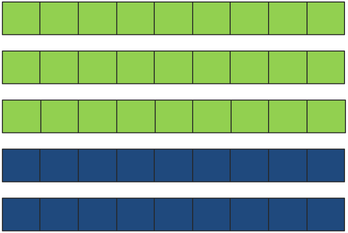
No of Computations: N+1

Evaluation of Rejection Conditions



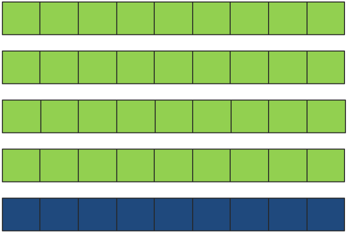
No of Computations: N+2

Evaluation of Rejection Conditions



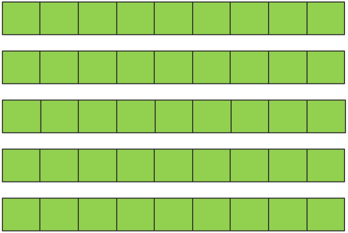
No of Computations: N+3

Evaluation of Rejection Conditions



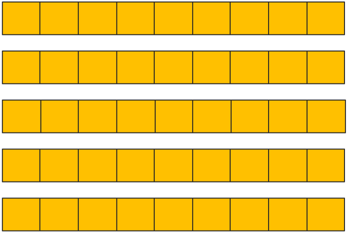
No of Computations: N+4

Evaluation of Rejection Conditions



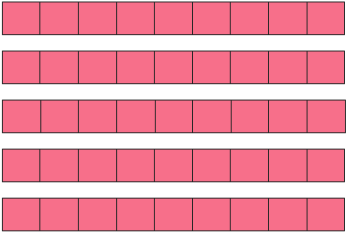
No of Computations: 2N

Evaluation of Rejection Conditions



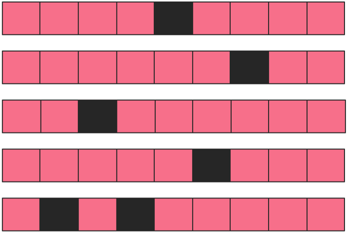
No of Computations: 3N

Evaluation of Rejection Conditions



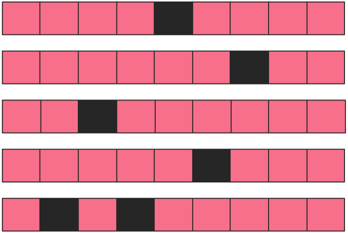
No of Computations: C*N

Evaluation of Rejection Conditions



Checking defective elements:

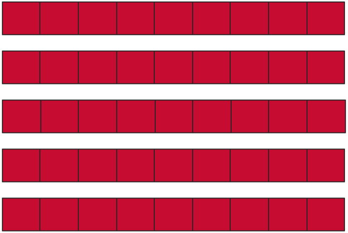
Evaluation of Rejection Conditions



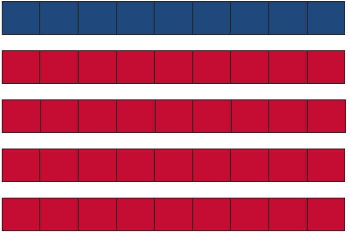
If (number of black boxes > 0)
Reject

Total Computations to compute rejection: condition **: (C+1)*N**

Early Evaluation of Rejection Conditions

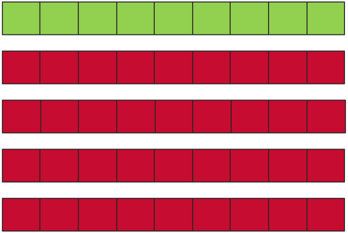


Early Evaluation of Rejection Conditions



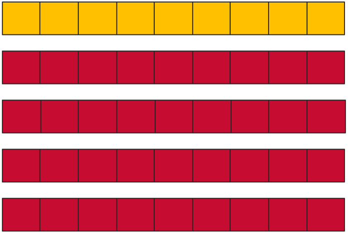
No of Computations: 1

Early Evaluation of Rejection Conditions



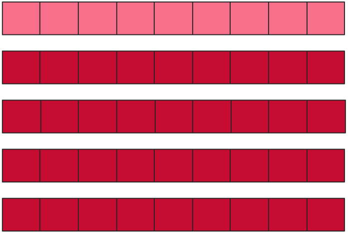
No of Computations: 2

Early Evaluation of Rejection Conditions



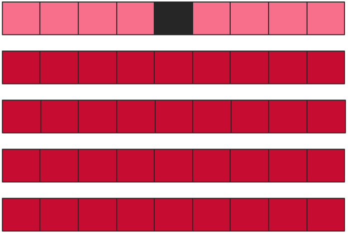
No of Computations: 3

Early Evaluation of Rejection Conditions



No of Computations: C

Early Evaluation of Rejection Conditions



No of Computations: C+1

Rejection done with only (C+1) computations

Early Evaluation of Rejection Conditions

- We perform the complete set of computations **one polynomial at a time**.
- **Best Case** - (C+1) computations.
- **Worst Case** - ((C+1)*N) computations.
- **Average Case** - ((C+1)*$\frac{N}{2}$) computations.
- We apply the same optimization to all the *Infy_Checks* in Dilithium's signing procedure.

Improving the Repetition Rate

Improving the Repetition Rate

- Total Repetition rate depends upon the failure rate of individual rejection conditions

Improving the Repetition Rate

- Total Repetition rate depends upon the failure rate of individual rejection conditions
- We specifically look at one rejection condition: $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$
- $\mathbf{z} = \mathbf{sc} + \mathbf{y}$.

Improving the Repetition Rate

- Total Repetition rate depends upon the failure rate of individual rejection conditions
- We specifically look at one rejection condition: $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$
- $\mathbf{z} = \mathbf{sc} + \mathbf{y}$.
- $\|\mathbf{y}\| \gg \|\mathbf{sc}\|$.

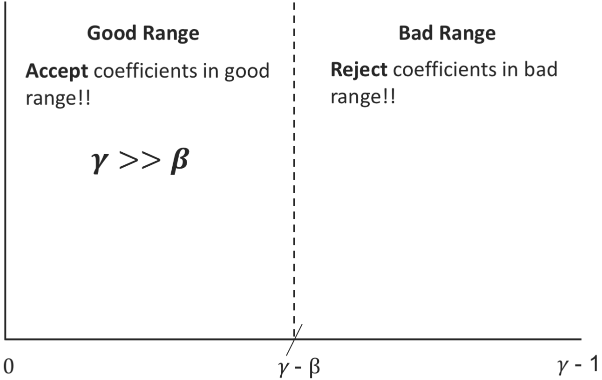
Improving the Repetition Rate

- Total Repetition rate depends upon the failure rate of individual rejection conditions
- We specifically look at one rejection condition: $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$
- $\mathbf{z} = \mathbf{sc} + \mathbf{y}$.
- $\|\mathbf{y}\| \gg \|\mathbf{sc}\|$.
- Coefficients of $\mathbf{y}$ are uniformly distributed in $[0, \gamma_1 - 1]$.

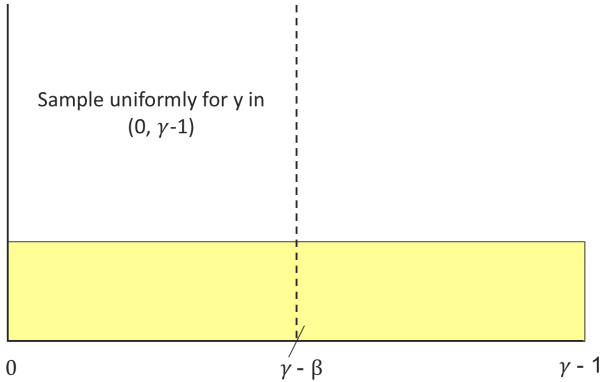
Improving the Repetition Rate

- Total Repetition rate depends upon the failure rate of individual rejection conditions
- We specifically look at one rejection condition: $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$
- $\mathbf{z} = \mathbf{sc} + \mathbf{y}$.
- $\|\mathbf{y}\| \gg \|\mathbf{sc}\|$.
- Coefficients of $\mathbf{y}$ are uniformly distributed in $[0, \gamma_1 - 1]$.
- Coefficients of $\mathbf{sc}$ are very small and normally distributed in $[0, \beta]$.

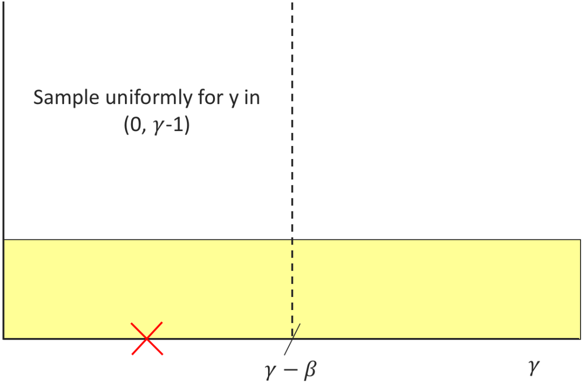
Generation of $\mathbf{z}$



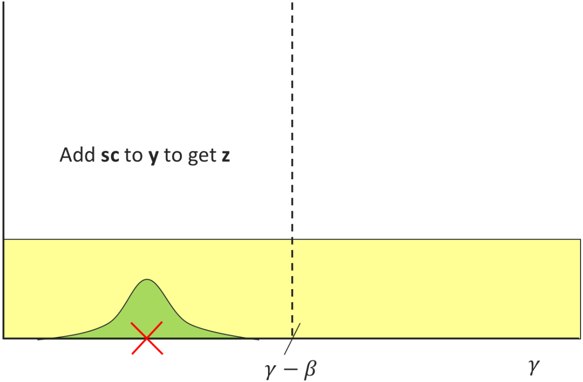
Generation of $\mathbf{z}$



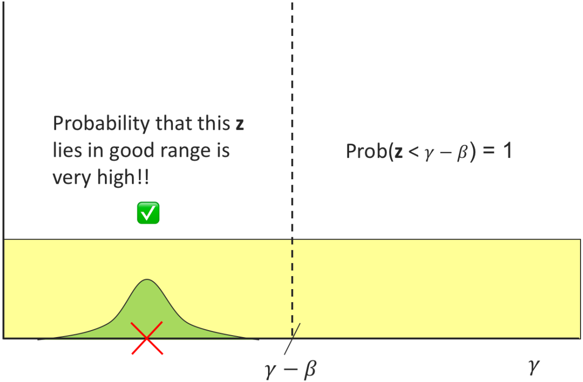
Generation of $\mathbf{z}$



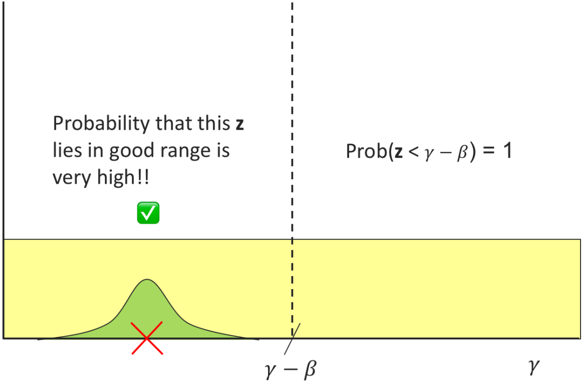
Generation of $\mathbf{z}$



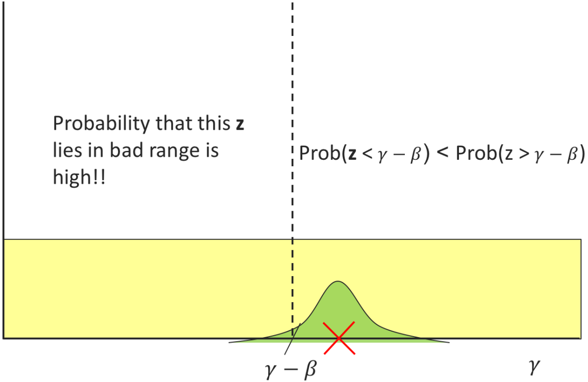
Generation of $\mathbf{z}$



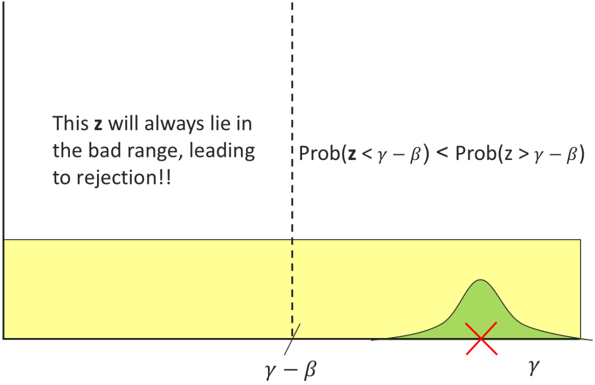
Generation of $\mathbf{z}$



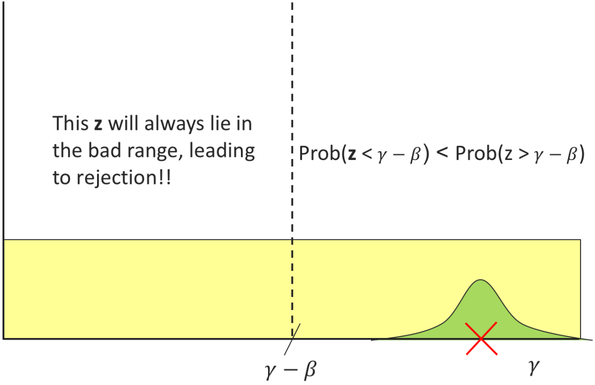
Generation of $\mathbf{z}$



Generation of $\mathbf{z}$



Generation of $\mathbf{z}$



Improving the Repetition Rate

- Rejection Sampling is performed so as to hide the $\mathbf{sc}$ component within $\mathbf{z}$.
- Allows to generate upto 2^{80} signatures without leaking the distribution of the $\mathbf{sc}$ component.

Improving the Repetition Rate

- Rejection Sampling is performed so as to hide the $\mathbf{sc}$ component within $\mathbf{z}$.
- Allows to generate upto 2^{80} signatures without leaking the distribution of the $\mathbf{sc}$ component.
- If $\mathbf{y} > \gamma_1 - \beta$, probability of $\mathbf{z}$ in bad range is very high.

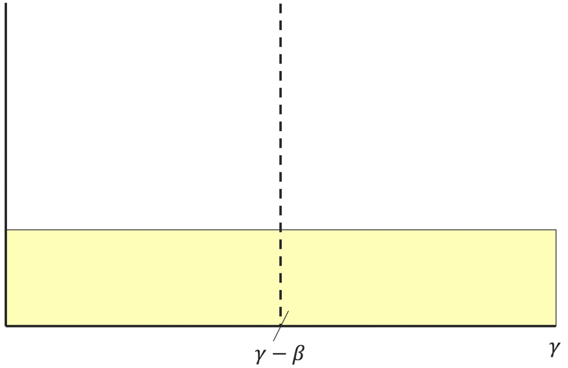
Improving the Repetition Rate

- Rejection Sampling is performed so as to hide the $\mathbf{sc}$ component within $\mathbf{z}$.
- Allows to generate upto 2^{80} signatures without leaking the distribution of the $\mathbf{sc}$ component.
- If $\mathbf{y} > \gamma_1 - \beta$, probability of $\mathbf{z}$ in bad range is very high.
- $\mathbf{y}$ is sampled uniformly in $[0, \gamma_1]$ and hence has a certain non-negligible probability that its corresponding $\mathbf{z}$ lies in the bad range.

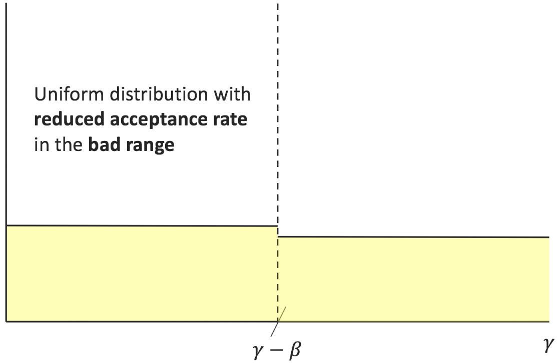
Improving the Repetition Rate

- Rejection Sampling is performed so as to hide the $\mathbf{sc}$ component within $\mathbf{z}$.
- Allows to generate upto 2^{80} signatures without leaking the distribution of the $\mathbf{sc}$ component.
- If $\mathbf{y} > \gamma_1 - \beta$, probability of $\mathbf{z}$ in bad range is very high.
- $\mathbf{y}$ is sampled uniformly in $[0, \gamma_1]$ and hence has a certain non-negligible probability that its corresponding $\mathbf{z}$ lies in the bad range.
- Can we alter the distribution of $\mathbf{y}$ so as to reduce the occurrence of $\mathbf{z}$ in the bad range?

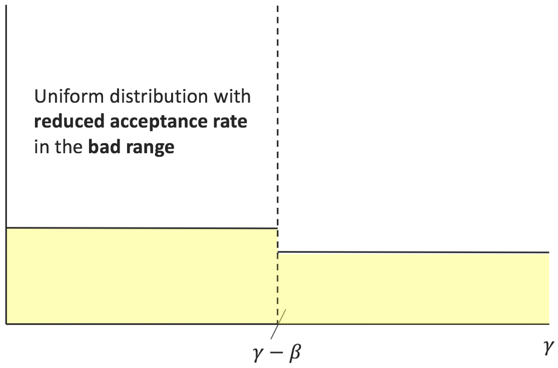
Uniform Distribution with Reduced Acceptance Rate



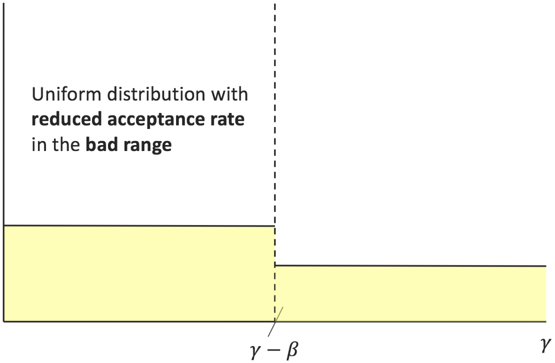
Uniform Distribution with Reduced Acceptance Rate



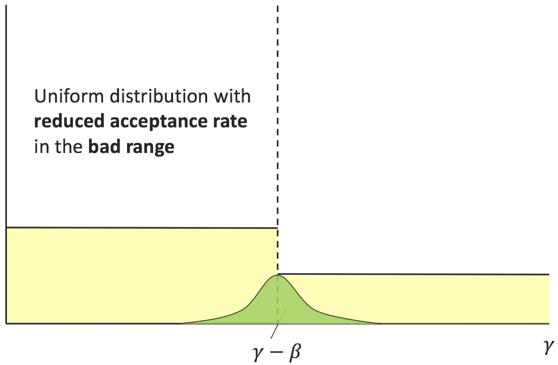
Uniform Distribution with Reduced Acceptance Rate



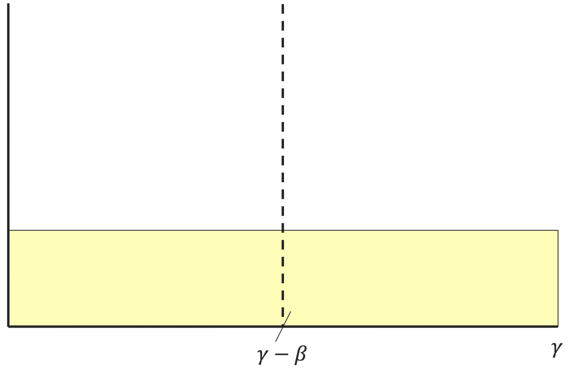
Uniform Distribution with Reduced Acceptance Rate



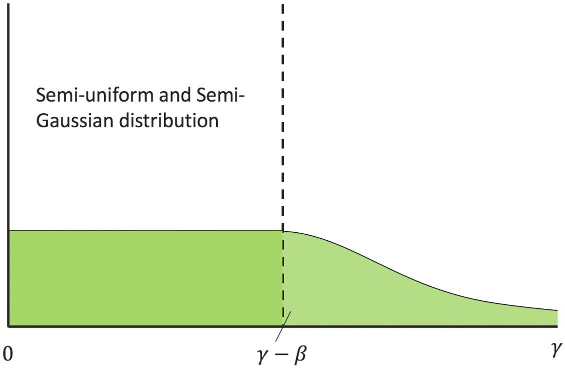
Uniform Distribution with Reduced Acceptance Rate



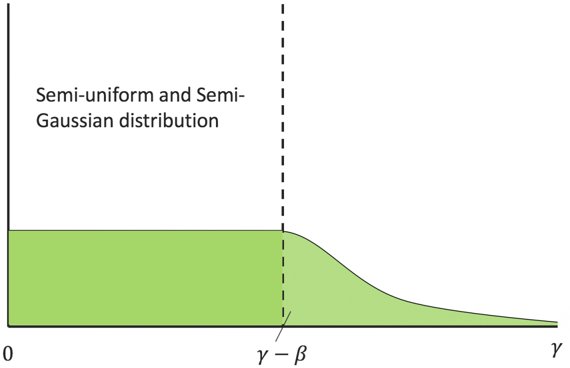
Semi-Uniform and Semi-Gaussian



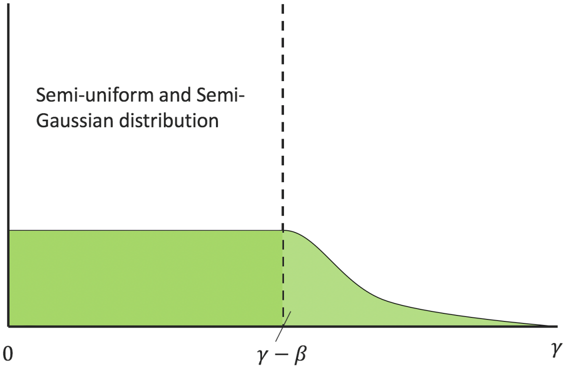
Semi-Uniform and Semi-Gaussian



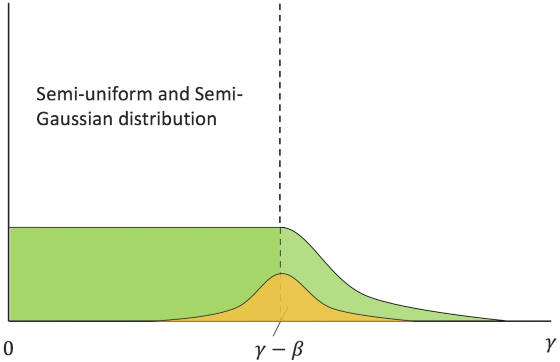
Semi-Uniform and Semi-Gaussian



Semi-Uniform and Semi-Gaussian



Semi-Uniform and Semi-Gaussian



Alternate Distributions for Sampling $\mathbf{y}$

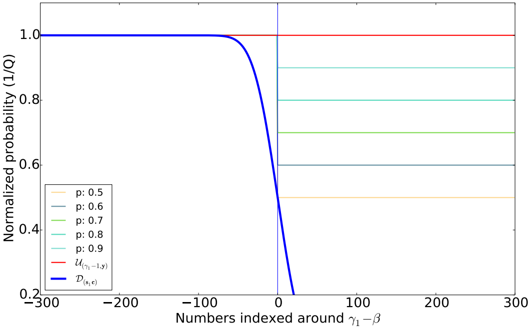


Figure: $\mathcal{U}_{(\gamma_1-\beta,\gamma_1-1,p)}$ - Uniform distribution with reduced acceptance rate p

Alternate Distributions for Sampling $\mathbf{y}$

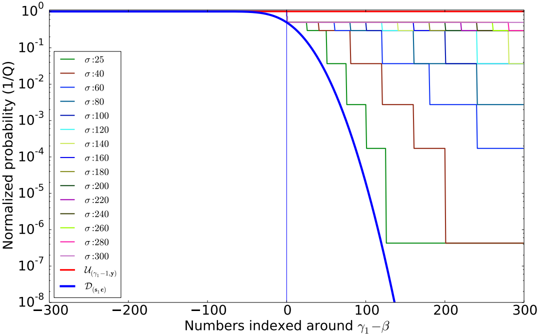


Figure: $\mathcal{D}_{(\gamma_1-\beta,\gamma_1-1,\sigma)}$ - Piece-wise Gaussian distribution with standard deviation σ

Table of Contents

1 Context

2 Background

3 Algorithmic Optimizations

4 Experimental Results

5 Future Work

6 Conclusion

Experimental Results

- Implementation of *Early-Eval* optimization and *Improved-Sampling* optimizations on reference implementation of `Dilithium`.
- Both the optimizations can be employed independently.
- Since both optimizations are done at the algorithmic level, they can be ported to all implementation platforms.
- Results were obtained for about 10^7 runs of the signing procedure.
- Implemented on Intel(R) Core(TM) i5-4460 CPU 3.20GHz and compiled with gcc-4.2.1.

Experimental Results

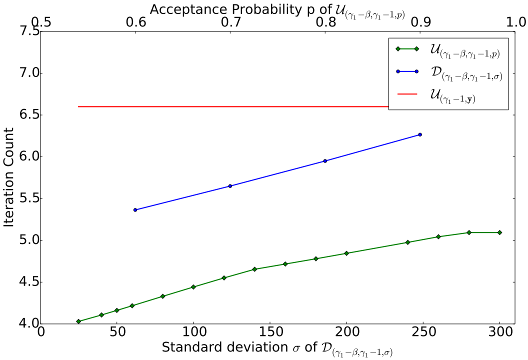


Figure: Improvements in iteration Count evaluated for various parameters of our alternate distributions

Experimental Results

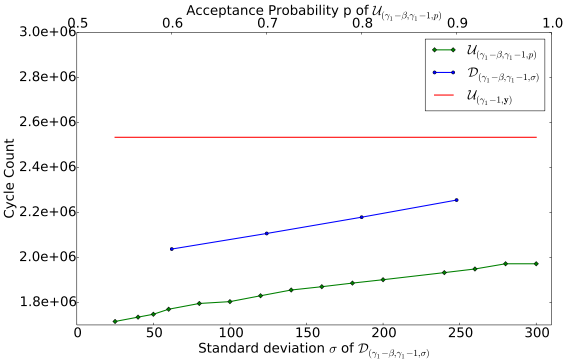


Figure: Improvements in Cycle count evaluated for various parameters of our alternate distributions

Experimental Results

- *Early-Eval* optimization yields improvement of about 8% in the signing speed.
- Combination of *Early-Eval* and *Improved-Sampling* optimizations could yield speed up upto 38%.
- *Early-Eval* optimization does not have any impact on security of the scheme.
- Does the use of improved distributions for $\mathbf{y}$ affect the security of the scheme? If so, by how much?
- How many signatures does the attacker need to observe an exploitable skew in the distribution of $\mathbf{z}$.
- This could lead to a potential quantitative trade-off between security and efficiency, which needs to be evaluated.

Table of Contents

1. Context
2. Background
3. Algorithmic Optimizations
4. Experimental Results
5. Future Work
6. Conclusion

Future Work

- Security Analysis of the signing procedure with improved distribution.
- Evaluation of the security-efficiency trade-off due to use of improved distributions.
- Utilization of a constant-time Gaussian sampler to sample from the improved distribution.

Table of Contents

1. Context
2. Background
3. Algorithmic Optimizations
4. Experimental Results
5. Future Work
6. Conclusion

Future Work

- This work proposes algorithmic optimizations for the `Dilithium` signature scheme
- We propose two optimizations:
  - *Early-Eval* optimization
  - *Improved-Sampling* optimization
- We were able to achieve a speed-up of upto 38% by employing a combination of both the optimizations.
- Incorporation of the *Improved-Sampling* optimization could lead to a potential security-efficiency trade-off.
- We intend to perform a quantitative evaluation of the security-efficiency trade-off as part of future work.

Thank you!

Any questions?