Lattice-based cryptography: Episode V: the ring strikes back

Daniel J. Bernstein
University of Illinois at Chicago

Crypto 1999 Nguyen: "At Crypto '97, Goldreich, Goldwasser and Halevi proposed a public-key cryptosystem based on the closest vector problem in a lattice, which is known to be NP-hard. We show that . . . the problem of decrypting ciphertexts can be

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

based cryptography:
V:
strikes back

. Bernstein
ty of Illinois at Chicago

1999 Nguyen: "At Crypto
dreich, Goldwasser and
roposed a public-key
stem based on the closest
roblem in a lattice, which
n to be NP-hard. We
at . . . the problem of
ng ciphertexts can be

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

Fix woul
dimensio
"Public

Crypto 1
"Provab
system b

tography:

ck

n
is at Chicago

en: "At Crypto
ldwasser and
public-key
d on the closest
a lattice, which
-hard. We
problem of
exts can be

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

Fix would "probab
dimension $\geq$ 400"
"Public key $\approx$ 1.8

Crypto 1998 Nguy
"Provably secure"
system breakable v

ago

rypto
nd
/
closest
which
e
of
e

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

Fix would "probably need dimension $\geq$ 400" for securi
"Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern:
"Provably secure" Ajtai–Dw
system breakable with 20MB

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

Fix would "probably need dimension $\geq$ 400" for security: "Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

reduced to a special closest vector problem which is much easier than the general problem. As an application, we solved four out of the five numerical challenges proposed on the Internet by the authors of the cryptosystem. At least two of those four challenges were conjectured to be intractable. We discuss ways to prevent the flaw, but conclude that, even modified, the scheme cannot provide sufficient security without being impractical."

Fix would "probably need dimension ≥ 400" for security: "Public key ≈ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

to a special closest vector
which is much easier
general problem. As an
ion, we solved four out
ve numerical challenges
d on the Internet by the
of the cryptosystem.
two of those four
es were conjectured to
ctable. We discuss ways
nt the flaw, but conclude
en modified, the scheme
provide sufficient security
being impractical."

Fix would "probably need dimension $\geq$ 400" for security: "Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05
following
"Lattice
"Lattice
currently
for post-

al closest vector
much easier
roblem. As an
lved four out
cal challenges
nternet by the
ptosystem.
ose four
njectured to
e discuss ways
v, but conclude
d, the scheme
fficient security
ractical."

Fix would "probably need dimension $\geq$ 400" for security: "Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05: Lattice s
following text to W
"Lattice-based cry
"Lattice-based con
currently the prim
for post-quantum

vector
er
As an
out
ges
the
.

to
ways
clude
eme
urity

Fix would "probably need dimension ≥ 400" for security: "Public key ≈ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05: Lattice student ad
following text to Wikipedia
"Lattice-based cryptography
"Lattice-based constructions
currently the primary candid
for post-quantum cryptograp

Fix would “probably need dimension $\geq$ 400” for security: “Public key $\approx$ 1.8 Mbytes”.

Crypto 1998 Nguyen–Stern: “Provably secure” Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05: Lattice student adds the following text to Wikipedia page “Lattice-based cryptography”: “Lattice-based constructions are currently the primary candidates for post-quantum cryptography.”

Fix would "probably need dimension $\geq$ 400" for security: "Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

Fix would "probably need dimension $\geq$ 400" for security: "Public key $\approx$ 1.8 Mbytes".

Crypto 1998 Nguyen–Stern: "Provably secure" Ajtai–Dwork system breakable with 20MB keys.

Compare to 1978 McEliece code-based cryptosystem: much more stable security story through dozens of attack papers. Typical parameters: 1MB key for $>2^{128}$ *post-quantum* security.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

ld "probably need
on $\geq$ 400" for security:
key $\approx$ 1.8 Mbytes".

1998 Nguyen–Stern:
ly secure" Ajtai–Dwork
breakable with 20MB keys.

e to 1978 McEliece
sed cryptosystem:
ore stable security story
dozens of attack papers.
parameters: 1MB key for
ost-quantum security.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

Google s
for publi

How car
work wit
Combine

1. Do *n*
large enc
connect
See, e.g.
Koblitz–

ly need
for security:
Mbytes".

en–Stern:
Ajtai–Dwork
with 20MB keys.

McEliece
system:
security story
attack papers.
s: 1MB key for
um security.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

Google sent only a
for public keys, cip

How can lattice-ba
work within a few
Combine two ingre

1. Do *not* take ke
large enough for th
connect to "well-s
See, e.g., 2016 Ch
Koblitz–Menezes–

ty:

ork
3 keys.

tory
pers.
ey for
y.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

Google sent only a few KB for public keys, ciphertexts.

How can lattice-based crypt
work within a few KB?
Combine two ingredients:

1. Do *not* take key sizes
large enough for theorems to
connect to "well-studied" SV
See, e.g., 2016 Chatterjee–
Koblitz–Menezes–Sarkar.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

Google sent only a few KB for public keys, ciphertexts.

How can lattice-based crypto work within a few KB? Combine two ingredients:

1. Do *not* take key sizes large enough for theorems to connect to "well-studied" SVP_γ. See, e.g., 2016 Chatterjee–Koblitz–Menezes–Sarkar.

2017.05: Lattice student adds the following text to Wikipedia page "Lattice-based cryptography": "Lattice-based constructions are currently the primary candidates for post-quantum cryptography."

— [citation needed]

2016.07: Google rolls out large-scale experiment with post-quantum crypto between Chrome and some Google sites. **Uses lattice-based crypto.**

Google sent only a few KB for public keys, ciphertexts.

How can lattice-based crypto work within a few KB? Combine two ingredients:

1. Do *not* take key sizes large enough for theorems to connect to "well-studied" SVP_γ. See, e.g., 2016 Chatterjee–Koblitz–Menezes–Sarkar.

2. **Use ideal lattices.** Hope that the extra structure doesn't damage security.

: Lattice student adds the
g text to Wikipedia page
-based cryptography":
-based constructions are
y the primary candidates
-quantum cryptography."

tion needed]

: Google rolls out
ale experiment with
antum crypto between
and some Google sites.
ttice-based crypto.

Google sent only a few KB
for public keys, ciphertexts.

How can lattice-based crypto
work within a few KB?
Combine two ingredients:

1. Do *not* take key sizes
large enough for theorems to
connect to "well-studied" SVP_γ.
See, e.g., 2016 Chatterjee–
Koblitz–Menezes–Sarkar.

2. **Use ideal lattices.**
Hope that the extra structure
doesn't damage security.

1996–19
Silverma

Define F
$\mathbf{Z}[x]/(x^5$

Element
$c_0 + c_1 x$
with inte

To multi
multiply
replace x
replace x
e.g.: $(x^1$
$= x^{300}$ –
$= 7x^{197}$

student adds the
Wikipedia page
ptography":
nstructions are
ary candidates
cryptography."

d]

olls out
nent with
pto between
Google sites.
ed crypto.

Google sent only a few KB
for public keys, ciphertexts.

How can lattice-based crypto
work within a few KB?
Combine two ingredients:

1. Do *not* take key sizes
large enough for theorems to
connect to "well-studied" SVP_γ.
See, e.g., 2016 Chatterjee–
Koblitz–Menezes–Sarkar.

2. **Use ideal lattices.**
Hope that the extra structure
doesn't damage security.

1996–1998 Hoffste
Silverman "NTRU

Define R as the ri
$\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are
$c_0 + c_1 x + c_2 x^2 +$
with integer coeffi

To multiply in R:
multiply polynomi
replace x^{503} with
replace x^{504} with
e.g.: $(x^{100} + x^{300}$
$= x^{300} + 8x^{500} +$
$= 7x^{197} + x^{300} +$

ds the

page

y":

s are

lates

phy."

en

tes.

Google sent only a few KB for public keys, ciphertexts.

How can lattice-based crypto work within a few KB?
Combine two ingredients:

1. Do *not* take key sizes large enough for theorems to connect to "well-studied" SVP_γ. See, e.g., 2016 Chatterjee–Koblitz–Menezes–Sarkar.

2. **Use ideal lattices.** Hope that the extra structure doesn't damage security.

1996–1998 Hoffstein–Pipher
Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomi
$c_0 + c_1 x + c_2 x^2 + \cdots + c_{502}$
with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R

Google sent only a few KB
for public keys, ciphertexts.

How can lattice-based crypto
work within a few KB?
Combine two ingredients:

1. Do *not* take key sizes
large enough for theorems to
connect to "well-studied" SVP_γ.
See, e.g., 2016 Chatterjee–Koblitz–Menezes–Sarkar.

2. **Use ideal lattices.**
Hope that the extra structure
doesn't damage security.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring
$\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomials
$c_0 + c_1 x + c_2 x^2 + \cdots + c_{502} x^{502}$
with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

sent only a few KB
c keys, ciphertexts.

n lattice-based crypto
thin a few KB?
e two ingredients:

ot take key sizes
ough for theorems to
to "well-studied" SVP_γ.
., 2016 Chatterjee–
-Menezes–Sarkar.

ideal lattices.
at the extra structure
damage security.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomials $c_0 + c_1 x + c_2 x^2 + \cdots + c_{502} x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define q

Alice's p
coefficie
This is 5

a few KB
phertexts.

ased crypto
KB?
edients:

y sizes
heorems to
tudied" SVP_γ.
atterjee–
Sarkar.

ces.
ra structure
ecurity.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomials $c_0 + c_1x + c_2x^2 + \cdots + c_{502}x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key:
coefficients in $\{0,$
This is $503 \cdot 11 =$

o

o

VP_γ.

re

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503}-1)$.

Elements of R are polynomials $c_0 + c_1x + c_2x^2 + \cdots + c_{502}x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100}+x^{300})(x^{200}+7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ wi
coefficients in $\{0, 1, \ldots, q -$
This is $503 \cdot 11 = 5533$ bits.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503}-1)$.

Elements of R are polynomials $c_0 + c_1x + c_2x^2 + \cdots + c_{502}x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100}+x^{300})(x^{200}+7x^{400})$
$= x^{300}+8x^{500}+7x^{700}$
$= 7x^{197}+x^{300}+8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$.
This is $503 \cdot 11 = 5533$ bits.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomials $c_0 + c_1 x + c_2 x^2 + \cdots + c_{502} x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q - 1\}$.
This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*:
e.g., all coefficients in $\{-1, 0, 1\}$.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503} - 1)$.

Elements of R are polynomials $c_0 + c_1 x + c_2 x^2 + \cdots + c_{502} x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100} + x^{300})(x^{200} + 7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$.
This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*:
e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$:
multiply A by b in R; add c;
reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

1996–1998 Hoffstein–Pipher–Silverman "NTRU":

Define R as the ring $\mathbf{Z}[x]/(x^{503}-1)$.

Elements of R are polynomials $c_0 + c_1x + c_2x^2 + \cdots + c_{502}x^{502}$ with integer coefficients c_j.

To multiply in R:
multiply polynomials;
replace x^{503} with 1;
replace x^{504} with x; etc.
e.g.: $(x^{100}+x^{300})(x^{200}+7x^{400})$
$= x^{300} + 8x^{500} + 7x^{700}$
$= 7x^{197} + x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$.
This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*:
e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$:
multiply A by b in R; add c;
reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

Bob sends $Ab + c \bmod q$.
This is also 5533 bits.

98 Hoffstein–Pipher–
n "NTRU":

R as the ring
$^{503} - 1)$.

s of R are polynomials
$x + c_2x^2 + \cdots + c_{502}x^{502}$
eger coefficients c_j.

iply in R:
polynomials;
x^{503} with 1;
x^{504} with x; etc.
$^{100} + x^{300})(x^{200} + 7x^{400})$
$+ 8x^{500} + 7x^{700}$
$+ x^{300} + 8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q - 1\}$. This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*: e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$: multiply A by b in R; add c; reduce each coefficient modulo q to the range $\{0, 1, \ldots, q - 1\}$.

Bob sends $Ab + c \bmod q$. This is also 5533 bits.

"Quotie
used in

Alice ge
for smal
(with su
i.e., dA

ein–Pipher–

":

ng

polynomials

$\cdots + c_{502}x^{502}$

cients c_j.

als;

1;

x; etc.

$)(x^{200} + 7x^{400})$

$7x^{700}$

$8x^{500}$ in R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$. This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*: e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$: multiply A by b in R; add c; reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

Bob sends $Ab + c \bmod q$. This is also 5533 bits.

"Quotient NTRU"

used in original N

Alice generated A

for small random

(with suitable inve

i.e., $dA - 3a$ mod

als

$_2x^{502}$

$x^{400})$

R.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q - 1\}$. This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*: e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$: multiply A by b in R; add c; reduce each coefficient modulo q to the range $\{0, 1, \ldots, q - 1\}$.

Bob sends $Ab + c \bmod q$. This is also 5533 bits.

"Quotient NTRU" (new nar
used in original NTRU desig

Alice generated $A = 3a/d$ ir
for small random a, d
(with suitable invertibility):
i.e., $dA - 3a \bmod q = 0$.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$.
This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*:
e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$:
multiply A by b in R; add c;
reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

Bob sends $Ab + c \bmod q$.
This is also 5533 bits.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q
for small random a, d
(with suitable invertibility):
i.e., $dA - 3a \bmod q = 0$.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$. This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*: e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$: multiply A by b in R; add c; reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

Bob sends $Ab + c \bmod q$. This is also 5533 bits.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Define $q = 2048$.

Alice's public key: $A \in R$ with coefficients in $\{0, 1, \ldots, q-1\}$. This is $503 \cdot 11 = 5533$ bits.

Bob generates random $b, c \in R$ *with small coefficients*: e.g., all coefficients in $\{-1, 0, 1\}$.

Bob computes $Ab + c \bmod q$: multiply A by b in R; add c; reduce each coefficient modulo q to the range $\{0, 1, \ldots, q-1\}$.

Bob sends $Ab + c \bmod q$. This is also 5533 bits.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

$q = 2048$.

public key: $A \in R$ with
nts in $\{0, 1, \ldots, q-1\}$.
$503 \cdot 11 = 5533$ bits.

erates random $b, c \in R$
all coefficients:
coefficients in $\{-1, 0, 1\}$.

nputes $Ab + c \bmod q$:
A by b in R; add c;
each coefficient modulo q
ange $\{0, 1, \ldots, q-1\}$.

ds $Ab + c \bmod q$.
lso 5533 bits.

"Quotient NTRU" (new name),
used in original NTRU design:

Alice generated $A = 3a/d$ in R/q
for small random a, d
(with suitable invertibility):
i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$.
Alice computes $dC \bmod q$,
i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$,
using smallness of a, b, d, c.
Alice computes dc,
deduces c, deduces b.

"Produc
2010 Ly

Everyone
Alice ge
for smal

$A \in R$ with
$1, \ldots, q-1\}$.
5533 bits.

dom $b, c \in R$
ents:
s in $\{-1, 0, 1\}$.

$+ c$ mod q:
R; add c;
cient modulo q
$, \ldots, q-1\}$.

mod q.
bits.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

"Product NTRU"
2010 Lyubashevsk

Everyone knows ra
Alice generated A
for small random

ith

1}.

$\in R$

$0, 1\}$.

q:

;

ulo q

}.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

"Product NTRU" (new nam

2010 Lyubashevsky–Peikert–

Everyone knows random $G \in$

Alice generated $A = aG + d$

for small random a, d.

“Quotient NTRU” (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

“Product NTRU” (new name), 2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$. Alice generated $A = aG + d \bmod q$ for small random a, d.

“Quotient NTRU” (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

“Product NTRU” (new name), 2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$. Alice generated $A = aG + d \bmod q$ for small random a, d.

Bob sends $B = Gb + e \bmod q$ and $C = m + Ab + c \bmod q$ where b, c, e are small and each coefficient of m is 0 or $q/2$.

"Quotient NTRU" (new name), used in original NTRU design:

Alice generated $A = 3a/d$ in R/q for small random a, d (with suitable invertibility): i.e., $dA - 3a \bmod q = 0$.

Alice receives $C = Ab + c \bmod q$. Alice computes $dC \bmod q$, i.e., $3ab + dc \bmod q$.

Alice reconstructs $3ab + dc$, using smallness of a, b, d, c. Alice computes dc, deduces c, deduces b.

"Product NTRU" (new name), 2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$. Alice generated $A = aG + d \bmod q$ for small random a, d.

Bob sends $B = Gb + e \bmod q$ and $C = m + Ab + c \bmod q$ where b, c, e are small and each coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$, i.e., $m + db + c - ae \bmod q$. Alice reconstructs m, using smallness of d, b, c, a, e.

nt NTRU" (new name),
original NTRU design:

nerated $A = 3a/d$ in R/q
l random a, d
itable invertibility):
$- 3a \bmod q = 0$.

ceives $C = Ab + c \bmod q$.
mputes $dC \bmod q$,
$+ dc \bmod q$.

constructs $3ab + dc$,
nallness of a, b, d, c.
mputes dc,
c, deduces b.

"Product NTRU" (new name),
2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$.
Alice generated $A = aG + d \bmod q$
for small random a, d.

Bob sends $B = Gb + e \bmod q$
and $C = m + Ab + c \bmod q$
where b, c, e are small and each
coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$,
i.e., $m + db + c - ae \bmod q$.
Alice reconstructs m,
using smallness of d, b, c, a, e.

Lattice v
the set o
such tha

(new name),
TRU design:

$= 3a/d$ in R/q
a, d
ertibility):
$q = 0$.

$Ab + c$ mod q.
C mod q,
d q.

$3ab + dc$,
a, b, d, c.
c,
s b.

"Product NTRU" (new name),
2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$.
Alice generated $A = aG + d$ mod q
for small random a, d.

Bob sends $B = Gb + e$ mod q
and $C = m + Ab + c$ mod q
where b, c, e are small and each
coefficient of m is 0 or $q/2$.

Alice computes $C - aB$ mod q,
i.e., $m + db + c - ae$ mod q.
Alice reconstructs m,
using smallness of d, b, c, a, e.

Lattice view: Defi
the set of pairs (v
such that $vG - w$

"Product NTRU" (new name),
2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$.
Alice generated $A = aG + d \bmod q$
for small random a, d.

Bob sends $B = Gb + e \bmod q$
and $C = m + Ab + c \bmod q$
where b, c, e are small and each
coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$,
i.e., $m + db + c - ae \bmod q$.
Alice reconstructs m,
using smallness of d, b, c, a, e.

Lattice view: Define L as
the set of pairs $(v, w) \in R \times$
such that $vG - w \bmod q =$

"Product NTRU" (new name), 2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$. Alice generated $A = aG + d \bmod q$ for small random a, d.

Bob sends $B = Gb + e \bmod q$ and $C = m + Ab + c \bmod q$ where b, c, e are small and each coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$, i.e., $m + db + c - ae \bmod q$. Alice reconstructs m, using smallness of d, b, c, a, e.

Lattice view: Define L as the set of pairs $(v, w) \in R \times R$ such that $vG - w \bmod q = 0$.

"Product NTRU" (new name),
2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$.
Alice generated $A = aG + d \bmod q$ for small random a, d.

Bob sends $B = Gb + e \bmod q$ and $C = m + Ab + c \bmod q$ where b, c, e are small and each coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$, i.e., $m + db + c - ae \bmod q$.
Alice reconstructs m, using smallness of d, b, c, a, e.

Lattice view: Define L as the set of pairs $(v, w) \in R \times R$ such that $vG - w \bmod q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU and Quotient NTRU.

"Product NTRU" (new name),
2010 Lyubashevsky–Peikert–Regev:

Everyone knows random $G \in R$.
Alice generated $A = aG + d \bmod q$ for small random a, d.

Bob sends $B = Gb + e \bmod q$ and $C = m + Ab + c \bmod q$ where b, c, e are small and each coefficient of m is 0 or $q/2$.

Alice computes $C - aB \bmod q$, i.e., $m + db + c - ae \bmod q$.
Alice reconstructs m, using smallness of d, b, c, a, e.

Lattice view: Define L as the set of pairs $(v, w) \in R \times R$ such that $vG - w \bmod q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU and Quotient NTRU.

Try to exploit reuse of b for faster Product NTRU attack.
("Ring-LWE": arbitrary reuse.)

Try to exploit $A = 3a/d$ structure for faster Quotient NTRU attack.

t NTRU" (new name),
ubashevsky–Peikert–Regev:

e knows random $G \in R$.
nerated $A = aG + d$ mod q
l random a, d.

ds $B = Gb + e$ mod q
$= m + Ab + c$ mod q
, c, e are small and each
nt of m is 0 or $q/2$.

mputes $C - aB$ mod q,
$- db + c - ae$ mod q.
onstructs m,
nallness of d, b, c, a, e.

Lattice view: Define L as
the set of pairs $(v, w) \in R \times R$
such that $vG - w$ mod $q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU
and Quotient NTRU.

Try to exploit reuse of b
for faster Product NTRU attack.
("Ring-LWE": arbitrary reuse.)

Try to exploit $A = 3a/d$ structure
for faster Quotient NTRU attack.

2013 Ly
Regev:
and algo
quantum
employ
to bear
problems
despite
significa
these pr
The bes
ideal lat
no bette
counterp
in practi

(new name),
y–Peikert–Regev:

andom $G \in R$.
$= aG + d \bmod q$
a, d.

$b + e \bmod q$
$+ c \bmod q$
mall and each
0 or $q/2$.

$- aB \bmod q$,
$- ae \bmod q$.
m,
d, b, c, a, e.

Lattice view: Define L as the set of pairs $(v, w) \in R \times R$ such that $vG - w \bmod q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU and Quotient NTRU.

Try to exploit reuse of b for faster Product NTRU attack.
("Ring-LWE": arbitrary reuse.)

Try to exploit $A = 3a/d$ structure for faster Quotient NTRU attack.

2013 Lyubashevsk
Regev: "All of the
and algorithmic to
quantum computa
employ . . . can als
to bear against SV
problems on ideal
despite considerab
significant progres
these problems ha
The best-known a
ideal lattices perfo
no better than the
counterparts, both
in practice."

ne),
-Regev:

$\in R$.
mod q

q

each

d q,
q.

e.

Lattice view: Define L as
the set of pairs $(v, w) \in R \times R$
such that $vG - w \bmod q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU
and Quotient NTRU.

Try to exploit reuse of b
for faster Product NTRU attack.
("Ring-LWE": arbitrary reuse.)

Try to exploit $A = 3a/d$ structure
for faster Quotient NTRU attack.

2013 Lyubashevsky–Peikert–
Regev: "All of the algebraic
and algorithmic tools (incluc
quantum computation) that
employ . . . can also be brou
to bear against SVP and oth
problems on ideal lattices. Y
despite considerable effort, r
significant progress in attack
these problems has been ma
The best-known algorithms
ideal lattices perform essenti
no better than their generic
counterparts, both in theory
in practice."

Lattice view: Define L as the set of pairs $(v, w) \in R \times R$ such that $vG - w \bmod q = 0$.

e.g. $(a, A - d) \in L$.
$(0, A)$ is close to a lattice point.

Try to find close lattice point.
Breaks both Product NTRU and Quotient NTRU.

Try to exploit reuse of b for faster Product NTRU attack.
(“Ring-LWE”: arbitrary reuse.)

Try to exploit $A = 3a/d$ structure for faster Quotient NTRU attack.

2013 Lyubashevsky–Peikert–Regev: “All of the algebraic and algorithmic tools (including quantum computation) that we employ . . . can also be brought to bear against SVP and other problems on ideal lattices. Yet despite considerable effort, no significant progress in attacking these problems has been made. The best-known algorithms for ideal lattices perform essentially no better than their generic counterparts, both in theory and in practice.”

view: Define L as
of pairs $(v, w) \in R \times R$
at $vG - w \bmod q = 0$.

$A - d) \in L$.
close to a lattice point.

nd close lattice point.
oth Product NTRU
otient NTRU.

xploit reuse of b
r Product NTRU attack.
LWE": arbitrary reuse.)

xploit $A = 3a/d$ structure
r Quotient NTRU attack.

2013 Lyubashevsky–Peikert–Regev: "All of the algebraic and algorithmic tools (including quantum computation) that we employ . . . can also be brought to bear against SVP and other problems on ideal lattices. Yet despite considerable effort, no significant progress in attacking these problems has been made. The best-known algorithms for ideal lattices perform essentially no better than their generic counterparts, both in theory and in practice."

Many m
(often n

Fully ho
STOC 2
"Fully h
using ide
PKC 20
Eurocryp
etc.

Multiline
Eurocryp
Halevi "
maps fro

ne L as

$, w) \in R \times R$

mod $q = 0$.

L.

lattice point.

attice point.

uct NTRU

RU.

se of b

NTRU attack.

itrary reuse.)

$= 3a/d$ structure

t NTRU attack.

2013 Lyubashevsky–Peikert–Regev: "All of the algebraic and algorithmic tools (including quantum computation) that we employ . . . can also be brought to bear against SVP and other problems on ideal lattices. Yet despite considerable effort, no significant progress in attacking these problems has been made. The best-known algorithms for ideal lattices perform essentially no better than their generic counterparts, both in theory and in practice."

Many more NTRU

(often not creditin

Fully homomorphi

STOC 2009 Gentr

"Fully homomorph

using ideal lattices

PKC 2010 Smart–

Eurocrypt 2011 Ge

etc.

Multilinear maps:

Eurocrypt 2013 Ga

Halevi "Candidate

maps from ideal la

× R
0.

oint.

nt.

tack.
e.)

ucture
ttack.

2013 Lyubashevsky–Peikert–Regev: “All of the algebraic and algorithmic tools (including quantum computation) that we employ . . . can also be brought to bear against SVP and other problems on ideal lattices. Yet despite considerable effort, no significant progress in attacking these problems has been made. The best-known algorithms for ideal lattices perform essentially no better than their generic counterparts, both in theory and in practice.”

Many more NTRU variants
(often not crediting NTRU).

Fully homomorphic encryptio
STOC 2009 Gentry
“Fully homomorphic encrypt
using ideal lattices”.
PKC 2010 Smart–Vercautere
Eurocrypt 2011 Gentry–Hale
etc.

Multilinear maps: e.g.,
Eurocrypt 2013 Garg–Gentry
Halevi “Candidate multilinea
maps from ideal lattices”.

2013 Lyubashevsky–Peikert–Regev: "All of the algebraic and algorithmic tools (including quantum computation) that we employ . . . can also be brought to bear against SVP and other problems on ideal lattices. Yet despite considerable effort, no significant progress in attacking these problems has been made. The best-known algorithms for ideal lattices perform essentially no better than their generic counterparts, both in theory and in practice."

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
"Fully homomorphic encryption using ideal lattices".
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g.,
Eurocrypt 2013 Garg–Gentry–Halevi "Candidate multilinear maps from ideal lattices".

ubashevsky–Peikert–
"All of the algebraic
orithmic tools (including
n computation) that we
. . . can also be brought
against SVP and other
s on ideal lattices. Yet
considerable effort, no
nt progress in attacking
oblems has been made.
t-known algorithms for
tices perform essentially
er than their generic
parts, both in theory and
ce."

Many more NTRU variants
(often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
"Fully homomorphic encryption
using ideal lattices".
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g.,
Eurocrypt 2013 Garg–Gentry–
Halevi "Candidate multilinear
maps from ideal lattices".

STOC 2
broken
for typic

y–Peikert–
e algebraic
ools (including
tion) that we
so be brought
/P and other
lattices. Yet
le effort, no
s in attacking
s been made.
lgorithms for
orm essentially
eir generic
n in theory and

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
"Fully homomorphic encryption using ideal lattices".
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g.,
Eurocrypt 2013 Garg–Gentry–Halevi "Candidate multilinear maps from ideal lattices".

STOC 2009 Gentr
broken by quantu
for typical "cycloto

ding
we
ght
her
Yet
no
king
de.
for
ially
and

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
“Fully homomorphic encryption using ideal lattices”.
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g.,
Eurocrypt 2013 Garg–Gentry–Halevi “Candidate multilinear maps from ideal lattices”.

STOC 2009 Gentry system i
broken by quantum algorith
for typical “cyclotomic rings

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
"Fully homomorphic encryption using ideal lattices".
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g., Eurocrypt 2013 Garg–Gentry–Halevi "Candidate multilinear maps from ideal lattices".

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption:
STOC 2009 Gentry
"Fully homomorphic encryption using ideal lattices".
PKC 2010 Smart–Vercauteren.
Eurocrypt 2011 Gentry–Halevi.
etc.

Multilinear maps: e.g., Eurocrypt 2013 Garg–Gentry–Halevi "Candidate multilinear maps from ideal lattices".

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack:
SODA 2016 Biasse–Song
fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Many more NTRU variants (often not crediting NTRU).

Fully homomorphic encryption: STOC 2009 Gentry "Fully homomorphic encryption using ideal lattices". PKC 2010 Smart–Vercauteren. Eurocrypt 2011 Gentry–Halevi. etc.

Multilinear maps: e.g., Eurocrypt 2013 Garg–Gentry–Halevi "Candidate multilinear maps from ideal lattices".

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

ore NTRU variants
ot crediting NTRU).

momorphic encryption:
009 Gentry
omomorphic encryption
eal lattices".
10 Smart–Vercauteren.
ot 2011 Gentry–Halevi.

ear maps: e.g.,
ot 2013 Garg–Gentry–
Candidate multilinear
om ideal lattices".

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second s
Campbe
fast pre-
for typic
to comp

J variants
g NTRU).

c encryption:
y
nic encryption
s".
-Vercauteren.
entry–Halevi.

e.g.,
arg–Gentry–
multilinear
attices".

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second stage of at
Campbell–Groves–
fast pre-quantum
for typical cycloto
to compute $ug \mapsto$

.

on:

tion

en.

evi.

y–

ar

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second stage of attack: 201 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

STOC 2009 Gentry system is **broken** by quantum algorithms for typical “cyclotomic rings”.

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

STOC 2009 Gentry system is **broken** by quantum algorithms for typical "cyclotomic rings".

First stage in attack: SODA 2016 Biasse–Song fast quantum algorithm to compute $gR \mapsto ug$ with $u \in R^*$.

Builds upon STOC 2014 Eisenträger–Hallgren–Kitaev–Song quantum $R \mapsto R^*$ algorithm.

Older pre-quantum algorithms take subexponential time.

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

These attacks exploit structure of cyclotomic rings. Rescue system by switching to another ring?

009 Gentry system is
by quantum algorithms
al "cyclotomic rings".

ge in attack:
016 Biasse–Song
ntum algorithm to
e $gR \mapsto ug$ with $u \in R^*$.

pon STOC 2014
ger–Hallgren–Kitaev–Song
n $R \mapsto R^*$ algorithm.

e-quantum algorithms
exponential time.

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

These attacks exploit structure of cyclotomic rings. Rescue system by switching to another ring?

2014.02
attack s
time for

Eurocry
Bernstei
Vredend
time pre
"multiqu

2016 Be
Lange–v
Prime":
Galois g
reduce a

y system is
m algorithms
omic rings".

ck:
e–Song
rithm to
g with $u \in R^*$.

C 2014
ren–Kitaev–Song
algorithm.

n algorithms
al time.

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

These attacks exploit structure of cyclotomic rings. Rescue system by switching to another ring?

2014.02 Bernstein
attack strategy; su
time for many cho

Eurocrypt 2017 Ba
Bernstein–de Vale
Vredendaal: quasi
time pre-quantum
"multiquadratic ri

2016 Bernstein–Ch
Lange–van Vreden
Prime": use prime
Galois group, inert
reduce attack surf

s
ms
”.

∈ R^*.

–Song
.

ms

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

These attacks exploit structure of cyclotomic rings. Rescue system by switching to another ring?

2014.02 Bernstein: pre-quan
attack strategy; subexponen
time for many choices of rin

Eurocrypt 2017 Bauch–
Bernstein–de Valence–Lange
Vredendaal: quasipolynomia
time pre-quantum attack fo
“multiquadratic rings”.

2016 Bernstein–Chuengsatia
Lange–van Vredendaal “NT
Prime”: use prime degree, l
Galois group, inert modulus;
reduce attack surface at low

Second stage of attack: 2014.10 Campbell–Groves–Shepherd fast pre-quantum algorithm for typical cyclotomic ring to compute $ug \mapsto$ short g.

Eurocrypt 2017 Cramer–Ducas–Wesolowski extension of CGS: for typical cyclotomic ring, find fairly short element of *any* ideal.

These attacks exploit structure of cyclotomic rings. Rescue system by switching to another ring?

2014.02 Bernstein: pre-quantum attack strategy; subexponential time for many choices of ring.

Eurocrypt 2017 Bauch–Bernstein–de Valence–Lange–van Vredendaal: quasipolynomial-time pre-quantum attack for "multiquadratic rings".

2016 Bernstein–Chuengsatiansup–Lange–van Vredendaal "NTRU Prime": use prime degree, large Galois group, inert modulus; reduce attack surface at low cost.