NTRU Prime

Daniel J. Bernstein

University of Illinois at Chicago &
Technische Universiteit Eindhoven

---

cr.yp.to/papers.html
#ntruprime is joint work with:

Chitchanok Chuengsatiansup
Tanja Lange
Christine van Vredendaal

Technische Universiteit Eindhoven

Focus of this talk: motivation.

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Prime

. Bernstein

ty of Illinois at Chicago &
che Universiteit Eindhoven

---

to/papers.html
rime is joint work with:

nok Chuengsatiansup
ange
e van Vredendaal

che Universiteit Eindhoven

---

f this talk: motivation.

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Imagine
"This is
The atta
not brea
MD5, so
point of
speculat
is contro
and crea
state of
alternati
the very

n

is at Chicago &
siteit Eindhoven

s.html
nt work with:

gsatiansup

lendaal

siteit Eindhoven

motivation.

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Imagine someone
"This is completel
The attack by Dol
not break any nor
MD5, so what exa
point of preventing
speculation about
is controversial an
and creates confus
state of the art. R
alternative hash fu
the very least quit

ago &
hoven

ith:

o

hoven

on.

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Imagine someone responding
"This is completely out of li
The attack by Dobbertin do
not break any normal usage
MD5, so what exactly is the
point of preventing it? This
speculation about MD5 colli
is controversial and non-scie
and creates confusion on the
state of the art. Recommen
alternative hash functions is
the very least quite prematu

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Can we predict future attacks?

1996 Dobbertin–Bosselaers–Preneel "RIPEMD-160: a strengthened version of RIPEMD": "It is anticipated that these techniques can be used to produce collisions for MD5 and perhaps also for RIPEMD. This will probably require an additional effort, but it no longer seems as far away as it was a year ago."

1996 Robshaw: Collisions "should be expected"; upgrade "when practical and convenient".

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Clearly not a real cryptographer. Maybe a standards organization.

predict future attacks?

bbertin–Bosselaers–
"RIPEMD-160:
thened version of
D": "It is anticipated that
chniques can be used to
collisions for MD5 and
also for RIPEMD. This
bably require an additional
ut it no longer seems as
as it was a year ago."

bshaw: Collisions "should
cted"; upgrade "when
and convenient".

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Clearly not a real cryptographer. Maybe a standards organization.

Now ima
saying th
are wors
cryptogr

ture attacks?

Bosselaers–
-160:
rsion of
anticipated that
can be used to
for MD5 and
IPEMD. This
ire an additional
nger seems as
a year ago."

ollisions "should
grade "when
enient".

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Clearly not a real cryptographer. Maybe a standards organization.

Now imagine a rel
saying that all of t
are worse than "pr
cryptographic hash

ks?

l that
d to
and
This
itional
s as
o."
should
en

Imagine someone responding:
"This is completely out of line.
The attack by Dobbertin does
not break any normal usage of
MD5, so what exactly is the
point of preventing it? This
speculation about MD5 collisions
is controversial and non-scientific,
and creates confusion on the
state of the art. Recommending
alternative hash functions is at
the very least quite premature."

Clearly not a real cryptographer.
Maybe a standards organization.

Now imagine a religious fana
saying that all of these func
are worse than "provably se
cryptographic hash functions

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Clearly not a real cryptographer. Maybe a standards organization.

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

Imagine someone responding: "This is completely out of line. The attack by Dobbertin does *not* break any normal usage of MD5, so what exactly is the point of preventing it? This speculation about MD5 collisions is controversial and non-scientific, and creates confusion on the state of the art. Recommending alternative hash functions is at the very least quite premature."

Clearly not a real cryptographer. Maybe a standards organization.

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

1991 "provably secure" example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured. Very easy security reduction: finding C collision implies computing a discrete logarithm.

someone responding:
completely out of line.
ack by Dobbertin does
k any normal usage of
what exactly is the
preventing it? This
ion about MD5 collisions
oversial and non-scientific,
tes confusion on the
the art. Recommending
ve hash functions is at
least quite premature.”

ot a real cryptographer.
standards organization.

Now imagine a religious fanatic saying that all of these functions are worse than “provably secure” cryptographic hash functions.

1991 “provably secure” example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured.
Very easy security reduction: finding C collision implies computing a discrete logarithm.

CvHP is
Horrible
Far wors
standard
compres

Security
1922 Kr
1986 Co
Schroep
1993 Go
1993 Sc
1994 Sh

responding:
y out of line.
bbertin does
mal usage of
ctly is the
g it? This
MD5 collisions
d non-scientific,
sion on the
Recommending
unctions is at
e premature."

cryptographer.
s organization.

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

1991 "provably secure" example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured.
Very easy security reduction: finding C collision implies computing a discrete logarithm.

CvHP is very bad
Horrible security fo
Far worse security
standard "unstruct
compression-funct

Security losses in
1922 Kraitchik (in
1986 Coppersmith
Schroeppel (NFS
1993 Gordon (gen
1993 Schirokauer
1994 Shor (quantu

g:
ne.
es
of
e

isions
ntific,
e
ding
at
re."

pher.
tion.

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

1991 "provably secure" example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured.
Very easy security reduction: finding C collision implies computing a discrete logarithm.

CvHP is very bad cryptograp
Horrible security for its spee
Far worse security record tha
standard "unstructured"
compression-function design

Security losses in C include
1922 Kraitchik (index calcul
1986 Coppersmith–Odlyzko–
Schroeppel (NFS predecesso
1993 Gordon (general DL N
1993 Schirokauer (faster NF
1994 Shor (quantum poly ti

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

1991 "provably secure" example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured.
Very easy security reduction: finding C collision implies computing a discrete logarithm.

CvHP is very bad cryptography.
Horrible security for its speed.
Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Now imagine a religious fanatic saying that all of these functions are worse than "provably secure" cryptographic hash functions.

1991 "provably secure" example, Chaum–van Heijst–Pfitzmann:

Choose p sensibly.
Define $C(x, y) = 4^x 9^y \bmod p$ for suitable ranges of x and y.

Simple, beautiful, structured.
Very easy security reduction: finding C collision implies computing a discrete logarithm.

CvHP is very bad cryptography.
Horrible security for its speed.
Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Imagine someone in 1991 saying "DL security is well understood".

agine a religious fanatic
hat all of these functions
e than "provably secure"
aphic hash functions.

rovably secure" example,
van Heijst–Pfitzmann:

p sensibly.
$C(x, y) = 4^x 9^y \bmod p$
ble ranges of x and y.

beautiful, structured.
sy security reduction:
C collision implies
ng a discrete logarithm.

CvHP is very bad cryptography.
Horrible security for its speed.
Far worse security record than
standard "unstructured"
compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–
Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Imagine someone in 1991 saying
"DL security is well understood".

We still
pre-quan
Which D

igious fanatic
these functions
rovably secure"
n functions.

cure" example,
–Pfitzmann:

$4^x 9^y$ mod p
of x and y.

structured.
reduction:
implies
ete logarithm.

CvHP is very bad cryptography. Horrible security for its speed. Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Imagine someone in 1991 saying "DL security is well understood".

We still use discre
pre-quantum publ
Which DL groups

atic
tions
cure"
s.

mple,
n:

p
y.

l.
:

hm.

CvHP is very bad cryptography.
Horrible security for its speed.
Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Imagine someone in 1991 saying "DL security is well understood".

We still use discrete logs for
pre-quantum public-key cryp
Which DL groups are best?

CvHP is very bad cryptography. Horrible security for its speed. Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include 1922 Kraitchik (index calculus); 1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor); 1993 Gordon (general DL NFS); 1993 Schirokauer (faster NFS); 1994 Shor (quantum poly time).

Imagine someone in 1991 saying "DL security is well understood".

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

CvHP is very bad cryptography. Horrible security for its speed. Far worse security record than standard “unstructured” compression-function designs.

Security losses in C include 1922 Kraitchik (index calculus); 1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor); 1993 Gordon (general DL NFS); 1993 Schirokauer (faster NFS); 1994 Shor (quantum poly time).

Imagine someone in 1991 saying “DL security is well understood”.

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus “is not likely to work on elliptic curves.”

CvHP is very bad cryptography. Horrible security for its speed. Far worse security record than standard "unstructured" compression-function designs.

Security losses in C include
1922 Kraitchik (index calculus);
1986 Coppersmith–Odlyzko–Schroeppel (NFS predecessor);
1993 Gordon (general DL NFS);
1993 Schirokauer (faster NFS);
1994 Shor (quantum poly time).

Imagine someone in 1991 saying "DL security is well understood".

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus "is not likely to work on elliptic curves."

1997 Rivest: "Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry."

very bad cryptography.
security for its speed.
se security record than
l "unstructured"
sion-function designs.

losses in C include
aitchik (index calculus);
ppersmith–Odlyzko–
pel (NFS predecessor);
rdon (general DL NFS);
hirokauer (faster NFS);
or (quantum poly time).

someone in 1991 saying
urity is well understood".

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus "is not likely to work on elliptic curves."

1997 Rivest: "Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry."

Are RSA
These sy
enabling
Many op
Attacks
>100 sc
Still mar

How ma
the state

cryptography.
or its speed.
record than
tured"
ion designs.

C include
dex calculus);
–Odlyzko–
predecessor);
eral DL NFS);
(faster NFS);
um poly time).

in 1991 saying
ell understood".

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus "is not likely to work on elliptic curves."

1997 Rivest: "Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry."

Are RSA, DSA, et
These systems hav
enabling attacks s
Many optimizatior
Attacks keep getti
>100 scientific pa
Still many unexplo

How many people
the state of the ar

phy.
d.
an

s.

us);
-
or);
FS);
FS);
me).

ying
ood”.

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus “is not likely to work on elliptic curves.”

1997 Rivest: “Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry.”

Are RSA, DSA, etc. less sca
These systems have structur
enabling attacks such as NF
Many optimization avenues.
Attacks keep getting better.
>100 scientific papers.
Still many unexplored avenu

How many people understan
the state of the art?

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus "is not likely to work on elliptic curves."

1997 Rivest: "Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry."

Are RSA, DSA, etc. less scary? These systems have structure enabling attacks such as NFS. Many optimization avenues. Attacks keep getting better. $>$100 scientific papers. Still many unexplored avenues.

How many people understand the state of the art?

We still use discrete logs for *pre-quantum public-key* crypto. Which DL groups are best?

1986 Miller proposes ECC. Gives detailed arguments that index calculus "is not likely to work on elliptic curves."

1997 Rivest: "Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry."

Are RSA, DSA, etc. less scary? These systems have structure enabling attacks such as NFS. Many optimization avenues. Attacks keep getting better. $>$100 scientific papers. Still many unexplored avenues.

How many people understand the state of the art?

Recurring themes in attacks: factorizations of ring elements; ring automorphisms; subfields; extending applicability (even to some curves!) via group maps.

use discrete logs for
ntum public-key crypto.
DL groups are best?

ller proposes ECC.
tailed arguments that
lculus "is not likely
on elliptic curves."

vest: "Over time, this
nge, but for now trying to
valuation of the security
iptic-curve cryptosystem
ike trying to get an
on of some recently
ed Chaldean poetry."

Are RSA, DSA, etc. less scary?
These systems have structure
enabling attacks such as NFS.
Many optimization avenues.
Attacks keep getting better.
$>$100 scientific papers.
Still many unexplored avenues.

How many people understand
the state of the art?

Recurring themes in attacks:
factorizations of ring elements;
ring automorphisms; subfields;
extending applicability (even to
some curves!) via group maps.

Which E

2005 Be
"have th
the num
for ellipt

2005 EC
"Some g
exist abo
attacks
recomme
fields."

te logs for
ic-key crypto.
are best?

ses ECC.
uments that
not likely
curves."

er time, this
for now trying to
of the security
e cryptosystem
to get an
e recently
an poetry."

Are RSA, DSA, etc. less scary?
These systems have structure
enabling attacks such as NFS.
Many optimization avenues.
Attacks keep getting better.
$>$100 scientific papers.
Still many unexplored avenues.

How many people understand
the state of the art?

Recurring themes in attacks:
factorizations of ring elements;
ring automorphisms; subfields;
extending applicability (even to
some curves!) via group maps.

Which ECC *fields*

2005 Bernstein: p
"have the virtue o
the number of sec
for elliptic-curve c

2005 ECRYPT key
"Some general co
exist about possib
attacks . . . As a f
recommend curves
fields." No extra a

pto.

at

is
ving to
urity
stem

”

Are RSA, DSA, etc. less scary?

These systems have structure
enabling attacks such as NFS.
Many optimization avenues.
Attacks keep getting better.
$>$100 scientific papers.
Still many unexplored avenues.

How many people understand
the state of the art?

Recurring themes in attacks:
factorizations of ring elements;
ring automorphisms; subfields;
extending applicability (even to
some curves!) via group maps.

Which ECC *fields* do we use

2005 Bernstein: prime fields
“have the virtue of minimizi
the number of security conc
for elliptic-curve cryptograph

2005 ECRYPT key-sizes rep
“Some general concerns
exist about possible future
attacks . . . As a first choice
recommend curves over prim
fields.” No extra automorph

Are RSA, DSA, etc. less scary?
These systems have structure enabling attacks such as NFS.
Many optimization avenues.
Attacks keep getting better.
$>$100 scientific papers.
Still many unexplored avenues.

How many people understand the state of the art?

Recurring themes in attacks: factorizations of ring elements; ring automorphisms; subfields; extending applicability (even to some curves!) via group maps.

Which ECC *fields* do we use?

2005 Bernstein: prime fields "have the virtue of minimizing the number of security concerns for elliptic-curve cryptography."

2005 ECRYPT key-sizes report: "Some general concerns exist about possible future attacks . . . As a first choice, we recommend curves over prime fields." No extra automorphisms.

Are RSA, DSA, etc. less scary?

These systems have structure enabling attacks such as NFS. Many optimization avenues. Attacks keep getting better. $>$100 scientific papers. Still many unexplored avenues.

How many people understand the state of the art?

Recurring themes in attacks: factorizations of ring elements; ring automorphisms; subfields; extending applicability (even to some curves!) via group maps.

Which ECC *fields* do we use?

2005 Bernstein: prime fields "have the virtue of minimizing the number of security concerns for elliptic-curve cryptography."

2005 ECRYPT key-sizes report: "Some general concerns exist about possible future attacks ... As a first choice, we recommend curves over prime fields." No extra automorphisms.

Imagine a response: "That's premature! $E(\mathbf{F}_{2^n})$ isn't broken!"

, DSA, etc. less scary?
ystems have structure
 attacks such as NFS.
ptimization avenues.
keep getting better.
ientific papers.
ny unexplored avenues.

ny people understand
e of the art?

g themes in attacks:
ntions of ring elements;
omorphisms; subfields;
g applicability (even to
rves!) via group maps.

Which ECC *fields* do we use?

2005 Bernstein: prime fields "have the virtue of minimizing the number of security concerns for elliptic-curve cryptography."

2005 ECRYPT key-sizes report: "Some general concerns exist about possible future attacks . . . As a first choice, we recommend curves over prime fields." No extra automorphisms.

Imagine a response: "That's premature! $E(\mathbf{F}_{2^n})$ isn't broken!"

Last exa
Halevi–F
"Candid
obfuscat
encrypti

UCLA p
to Sahai
techniqu
presente
forcing a
effort, p
to revers
The new
an 'iron
in the fie

c. less scary?
ve structure
uch as NFS.
n avenues.
ng better.
pers.
ored avenues.

understand
t?

in attacks:
ng elements;
ns; subfields;
bility (even to
group maps.

Which ECC *fields* do we use?

2005 Bernstein: prime fields “have the virtue of minimizing the number of security concerns for elliptic-curve cryptography.”

2005 ECRYPT key-sizes report: “Some general concerns exist about possible future attacks . . . As a first choice, we recommend curves over prime fields.” No extra automorphisms.

Imagine a response: “That’s premature! $E(\mathbf{F}_{2^n})$ isn’t broken!”

Last example: 201
Halevi–Raykova–S
“Candidate indisti
obfuscation and fu
encryption for all c

UCLA press releas
to Sahai, previousl
techniques for obfu
presented only a ‘s
forcing an attacke
effort, perhaps a fe
to reverse-enginee
The new system, h
an ‘iron wall’ . . . a
in the field of cryp

ry?
re
S.

es.

nd

:
ts;
ds;
to
ps.

Which ECC *fields* do we use?

2005 Bernstein: prime fields "have the virtue of minimizing the number of security concerns for elliptic-curve cryptography."

2005 ECRYPT key-sizes report: "Some general concerns exist about possible future attacks . . . As a first choice, we recommend curves over prime fields." No extra automorphisms.

Imagine a response: "That's premature! $E(\mathbf{F}_{2^n})$ isn't broken!"

Last example: 2013 Garg–G
Halevi–Raykova–Sahai–Wate
"Candidate indistinguishabil
obfuscation and functional
encryption for all circuits".

UCLA press release: "Accor
to Sahai, previously develop
techniques for obfuscation
presented only a 'speed bum
forcing an attacker to spend
effort, perhaps a few days, t
to reverse-engineer the softw
The new system, he said, pu
an 'iron wall' . . . a game-ch
in the field of cryptography."

Which ECC *fields* do we use?

2005 Bernstein: prime fields "have the virtue of minimizing the number of security concerns for elliptic-curve cryptography."

2005 ECRYPT key-sizes report: "Some general concerns exist about possible future attacks . . . As a first choice, we recommend curves over prime fields." No extra automorphisms.

Imagine a response: "That's premature! $E(\mathbf{F}_{2^n})$ isn't broken!"

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

ECC *fields* do we use?

rnstein: prime fields
ne virtue of minimizing
ber of security concerns
ic-curve cryptography."

CRYPT key-sizes report:
general concerns
out possible future
. . . As a first choice, we
end curves over prime
No extra automorphisms.

a response: "That's
re! $E(\mathbf{F}_{2^n})$ isn't broken!"

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

2013 Be
cryptogr
of this s
the best
has agai
Security

do we use?

rime fields
f minimizing
urity concerns
ryptography."

y-sizes report:
ncerns
le future
irst choice, we
s over prime
automorphisms.

e: "That's
) isn't broken!"

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

2013 Bernstein: "
cryptographic con
of this sort of shit
the best defense t
has against the U.
Security Agency, w

e?
g
ng
erns
ny."
ort:
, we
ne
nisms.
s
ken!"

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

2013 Bernstein: "The flagsh
cryptographic conferences ar
of this sort of shit, and, if th
the best defense that the wo
has against the U.S. Nationa
Security Agency, we're screw

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

Last example: 2013 Garg–Gentry–Halevi–Raykova–Sahai–Waters "Candidate indistinguishability obfuscation and functional encryption for all circuits".

UCLA press release: "According to Sahai, previously developed techniques for obfuscation presented only a 'speed bump,' forcing an attacker to spend some effort, perhaps a few days, trying to reverse-engineer the software. The new system, he said, puts up an 'iron wall' . . . a game-change in the field of cryptography."

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

mple: 2013 Garg–Gentry–
Raykova–Sahai–Waters
ate indistinguishability
tion and functional
on for all circuits".

ress release: "According
, previously developed
ies for obfuscation
d only a 'speed bump,'
an attacker to spend some
erhaps a few days, trying
se-engineer the software.
system, he said, puts up
wall' . . . a game-change
eld of cryptography."

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic N

Standard
Also sta
Define $\mathcal{R}$

Receiver
(Some in
Public k

Sender c
Ciphert

.3 Garg–Gentry–
ahai–Waters
nguishability
unctional
circuits".

e: "According
ly developed
uscation
speed bump,'
r to spend some
ew days, trying
r the software.
he said, puts up
a game-change
ptography."

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime
Also standardize q
Define $\mathcal{R} = \mathbf{Z}[x]/$

Receiver chooses s
(Some invertibility
Public key $h = 3g$

Sender chooses sm
Ciphertext $c = m$

entry–
ers
ity

ding
ed

np,'
some
rying
vare.
its up
ange
''

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 74

Also standardize q; e.g. 204

Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small f, g

(Some invertibility requireme

Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in$

Ciphertext $c = m + hr \bmod$

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by $f \bmod q$: $fc \bmod q$.

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q: $fc \bmod q$.
Use smallness: $fm + 3gr$.

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q: $fc \bmod q$.
Use smallness: $fm + 3gr$.
Reduce mod 3: $fm \bmod 3$.

2013 Bernstein: "The flagship cryptographic conferences are full of this sort of shit, and, if this is the best defense that the world has against the U.S. National Security Agency, we're screwed."

2016 Miles–Sahai–Zhandry: "We exhibit two simple programs that are functionally equivalent, and show how to efficiently distinguish between the obfuscations of these two programs."

So Sahai's claimed "iron wall" is just another "speed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q: $fc \bmod q$.
Use smallness: $fm + 3gr$.
Reduce mod 3: $fm \bmod 3$.
Divide by f mod 3: m.

rnstein: “The flagship
aphic conferences are full
ort of shit, and, if this is
defense that the world
nst the U.S. National
Agency, we’re screwed.”

les–Sahai–Zhandry: “We
wo simple programs that
tionally equivalent, and
w to efficiently distinguish
the obfuscations
two programs.”

i’s claimed “iron wall”
nother “speed bump”.

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q: $fc \bmod q$.
Use smallness: $fm + 3gr$.
Reduce mod 3: $fm \bmod 3$.
Divide by f mod 3: m.

1998 Ho
introduc

Many su
meet-in-
lattice a
chosen-c
decrypti
complica
variation
paramet

Also ma
were sm
e.g., hor

The flagship
ferences are full
, and, if this is
hat the world
S. National
ve're screwed."

-Zhandry: "We
programs that
uivalent, and
ently distinguish
cations
ams."

d "iron wall"
beed bump".

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q:	$fc \bmod q$.
Use smallness:	$fm + 3gr$.
Reduce mod 3:	$fm \bmod 3$.
Divide by f mod 3:	m.

1998 Hoffstein–Pi
introduced this sys

Many subsequent
meet-in-the-middle
lattice attacks, hy
chosen-ciphertext
decryption-failure
complicated paddi
variations for effici
parameter selectio

Also many ideas t
were small tweaks
e.g., homomorphic

hip
re full
his is
orld
al
ved.”

“We
that
and
nguish

ll”
s”.

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q: $fc \bmod q$.
Use smallness: $fm + 3gr$.
Reduce mod 3: $fm \bmod 3$.
Divide by f mod 3: m.

1998 Hoffstein–Pipher–Silve
introduced this system.

Many subsequent NTRU pa
meet-in-the-middle attacks,
lattice attacks, hybrid attac
chosen-ciphertext attacks;
decryption-failure attacks;
complicated padding system
variations for efficiency;
parameter selection.

Also many ideas that in retr
were small tweaks of NTRU
e.g., homomorphic encryptio

Classic NTRU

Standardize prime p; e.g. 743.
Also standardize q; e.g. 2048.
Define $\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

Receiver chooses small $f, g \in \mathcal{R}$.
(Some invertibility requirements.)
Public key $h = 3g/f \bmod q$.

Sender chooses small $m, r \in \mathcal{R}$.
Ciphertext $c = m + hr \bmod q$.

Multiply by f mod q:	$fc \bmod q$.
Use smallness:	$fm + 3gr$.
Reduce mod 3:	$fm \bmod 3$.
Divide by f mod 3:	m.

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers:
meet-in-the-middle attacks,
lattice attacks, hybrid attacks;
chosen-ciphertext attacks;
decryption-failure attacks;
complicated padding systems;
variations for efficiency;
parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU:
e.g., homomorphic encryption.

NTRU

dize prime p; e.g. 743.

ndardize q; e.g. 2048.

$\mathcal{R} = \mathbf{Z}[x]/(x^p - 1)$.

chooses small $f, g \in \mathcal{R}$.

nvertibility requirements.)

ey $h = 3g/f \bmod q$.

chooses small $m, r \in \mathcal{R}$.

xt $c = m + hr \bmod q$.

by f mod q: $fc \bmod q$.

llness: $fm + 3gr$.

mod 3: $fm \bmod 3$.

y f mod 3: m.

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers:
meet-in-the-middle attacks,
lattice attacks, hybrid attacks;
chosen-ciphertext attacks;
decryption-failure attacks;
complicated padding systems;
variations for efficiency;
parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU: e.g., homomorphic encryption.

Unneces

Attacker
public p

Compati
multiplic
$f(1)h(1)$
$c(1) =$

p; e.g. 743.

; e.g. 2048.

$(x^p - 1)$.

small $f, g \in \mathcal{R}$.

requirements.)

$/f \bmod q$.

nall $m, r \in \mathcal{R}$.

$+ hr \bmod q$.

d q: $fc \bmod q$.

$fm + 3gr$.

$fm \bmod 3$.

: m.

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers: meet-in-the-middle attacks, lattice attacks, hybrid attacks; chosen-ciphertext attacks; decryption-failure attacks; complicated padding systems; variations for efficiency; parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU: e.g., homomorphic encryption.

Unnecessary struct

Attacker can evalu
public polynomials

Compatible with a
multiplication mod
$f(1)h(1) = 3g(1)$
$c(1) = m(1) + h(1$

-3.
8.

$\in \mathcal{R}$.
ents.)
.

$\mathcal{R}$.
q.

mod q.
$+ 3gr$.
mod 3.
m.

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers: meet-in-the-middle attacks, lattice attacks, hybrid attacks; chosen-ciphertext attacks; decryption-failure attacks; complicated padding systems; variations for efficiency; parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU: e.g., homomorphic encryption.

Unnecessary structures in N

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition a
multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}$

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers: meet-in-the-middle attacks, lattice attacks, hybrid attacks; chosen-ciphertext attacks; decryption-failure attacks; complicated padding systems; variations for efficiency; parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU: e.g., homomorphic encryption.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

1998 Hoffstein–Pipher–Silverman introduced this system.

Many subsequent NTRU papers: meet-in-the-middle attacks, lattice attacks, hybrid attacks; chosen-ciphertext attacks; decryption-failure attacks; complicated padding systems; variations for efficiency; parameter selection.

Also many ideas that in retrospect were small tweaks of NTRU: e.g., homomorphic encryption.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is guessable, sometimes standard.
Attacker scans many ciphertexts to find some with large $m(1)$.
Uses this to speed up m search.

offstein–Pipher–Silverman
ed this system.

ubsequent NTRU papers:
the-middle attacks,
ttacks, hybrid attacks;
iphertext attacks;
on-failure attacks;
ated padding systems;
ns for efficiency;
er selection.

ny ideas that in retrospect
all tweaks of NTRU:
momorphic encryption.

Unnecessary structures in NTRU

Attacker can evaluate
public polynomials h, c at 1.

Compatible with addition and
multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is
guessable, sometimes standard.
Attacker scans many ciphertexts
to find some with large $m(1)$.
Uses this to speed up m search.

NTRU c
so that
Limits in

pher–Silverman
stem.

NTRU papers:
e attacks,
brid attacks;
attacks;
attacks;
ng systems;
iency;
n.

hat in retrospect
of NTRU:
c encryption.

Unnecessary structures in NTRU

Attacker can evaluate
public polynomials h, c at 1.

Compatible with addition and
multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is
guessable, sometimes standard.
Attacker scans many ciphertexts
to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates
so that $m(1)$ is ne
Limits impact of t

rman

pers:

ks;

s;

ospect

:

on.

Unnecessary structures in NTRU

Attacker can evaluate
public polynomials h, c at 1.

Compatible with addition and
multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is
guessable, sometimes standard.
Attacker scans many ciphertexts
to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates m select
so that $m(1)$ is never large.
Limits impact of the attack.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is guessable, sometimes standard.
Attacker scans many ciphertexts to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates m selection so that $m(1)$ is never large.
Limits impact of the attack.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is guessable, sometimes standard.
Attacker scans many ciphertexts to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates m selection so that $m(1)$ is never large.
Limits impact of the attack.

Better: replace NTRU's
$\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$.
Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is guessable, sometimes standard.
Attacker scans many ciphertexts to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates m selection so that $m(1)$ is never large.
Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$.
Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p.
Compatible with add, mult.
Why include $m(1)$ here?
Doesn't seem to help security.

Unnecessary structures in NTRU

Attacker can evaluate public polynomials h, c at 1.

Compatible with addition and multiplication mod $x^p - 1$:
$f(1)h(1) = 3g(1)$ in $\mathbf{Z}/q$;
$c(1) = m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

One way to exploit this:
$c(1), h(1)$ are visible; $r(1)$ is guessable, sometimes standard.
Attacker scans many ciphertexts to find some with large $m(1)$.
Uses this to speed up m search.

NTRU complicates m selection so that $m(1)$ is never large.
Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$.
Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p.
Compatible with add, mult.
Why include $m(1)$ here?
Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

sary structures in NTRU

r can evaluate
olynomials h, c at 1.

ible with addition and
cation mod $x^p - 1$:
$) = 3g(1)$ in $\mathbf{Z}/q$;
$m(1) + h(1)r(1)$ in $\mathbf{Z}/q$.

y to exploit this:
1) are visible; $r(1)$ is
e, sometimes standard.
r scans many ciphertexts
some with large $m(1)$.
s to speed up m search.

NTRU complicates m selection so that $m(1)$ is never large. Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$. Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p. Compatible with add, mult. Why include $m(1)$ here? Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

More ge
any ring
to the e
and $c =$

tures in NTRU

uate
s h, c at 1.

ddition and
d $x^p - 1$:
in $\mathbf{Z}/q$;
1)$r(1)$ in $\mathbf{Z}/q$.

t this:
le; $r(1)$ is
mes standard.
ny ciphertexts
large $m(1)$.
up m search.

NTRU complicates m selection so that $m(1)$ is never large. Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$. Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p. Compatible with add, mult. Why include $m(1)$ here? Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

More generally: A
any ring map $(\mathbf{Z}/$
to the equations h
and $c = m + hr$ in

TRU

nd

$\mathbf{Z}/q$.

ard.

exts

).

rch.

NTRU complicates m selection so that $m(1)$ is never large. Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$. Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p. Compatible with add, mult. Why include $m(1)$ here? Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

More generally: Attacker ap any ring map $(\mathbf{Z}/q)[x]/P \to$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]$

NTRU complicates m selection so that $m(1)$ is never large. Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$. Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p. Compatible with add, mult. Why include $m(1)$ here? Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

NTRU complicates m selection so that $m(1)$ is never large. Limits impact of the attack.

Better: replace NTRU's $\mathbf{Z}[x]/(x^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$. Recall $\Phi_p = (x^p - 1)/(x - 1)$.

Can view poly m mod $x^p - 1$ as two parts: $m(1)$; m mod Φ_p. Compatible with add, mult. Why include $m(1)$ here? Doesn't seem to help security.

Or use other irreds. Ring-LWE typically uses $\Phi_{2048} = x^{1024} + 1$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU. Have natural ring maps from $(\mathbf{Z}/2048)[x]/(x^p - 1)$ to
$(\mathbf{Z}/2)[x]/(x^p - 1)$,
$(\mathbf{Z}/4)[x]/(x^p - 1)$,
$(\mathbf{Z}/8)[x]/(x^p - 1)$, etc.
Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

complicates m selection
$m(1)$ is never large.
mpact of the attack.

replace NTRU's
$^p - 1)$ with $\mathbf{Z}[x]/\Phi_p$.
$_p = (x^p - 1)/(x - 1)$.

v poly m mod $x^p - 1$
parts: $m(1)$; m mod Φ_p.
ible with add, mult.
lude $m(1)$ here?
seem to help security.

other irreds. Ring-LWE
uses $\Phi_{2048} = x^{1024} + 1$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU.
Have natural ring maps from
$(\mathbf{Z}/2048)[x]/(x^p - 1)$ to
$(\mathbf{Z}/2)[x]/(x^p - 1)$,
$(\mathbf{Z}/4)[x]/(x^p - 1)$,
$(\mathbf{Z}/8)[x]/(x^p - 1)$, etc.
Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

Ring-LW
"provabl
q so tha
$\mathbf{Z}[x]/q$;
maps $(\mathbf{Z}$

s m selection
ver large.
he attack.

TRU's
th $\mathbf{Z}[x]/\Phi_p$.
$1)/(x-1)$.

mod $x^p - 1$
); m mod Φ_p.
dd, mult.
here?
elp security.

s. Ring-LWE
$_{48} = x^{1024} + 1$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU. Have natural ring maps from $(\mathbf{Z}/2048)[x]/(x^p - 1)$ to $(\mathbf{Z}/2)[x]/(x^p - 1)$, $(\mathbf{Z}/4)[x]/(x^p - 1)$, $(\mathbf{Z}/8)[x]/(x^p - 1)$, etc. Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

Ring-LWE religion
"provable security"
q so that P splits
$\mathbf{Z}[x]/q$; i.e., have
maps $(\mathbf{Z}/q)[x]/P$

ion

p.

1).

1

Φ_p.

ty.

VE

$+1$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU. Have natural ring maps from $(\mathbf{Z}/2048)[x]/(x^p - 1)$ to
$(\mathbf{Z}/2)[x]/(x^p - 1)$,
$(\mathbf{Z}/4)[x]/(x^p - 1)$,
$(\mathbf{Z}/8)[x]/(x^p - 1)$, etc.
Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

Ring-LWE religion, version 1
"provable security", take pri
q so that P splits completel
$\mathbf{Z}[x]/q$; i.e., have n different
maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU. Have natural ring maps from $(\mathbf{Z}/2048)[x]/(x^p - 1)$ to $(\mathbf{Z}/2)[x]/(x^p - 1)$, $(\mathbf{Z}/4)[x]/(x^p - 1)$, $(\mathbf{Z}/8)[x]/(x^p - 1)$, etc. Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

More generally: Attacker applies any ring map $(\mathbf{Z}/q)[x]/P \to T$ to the equations $h = 3g/f$ and $c = m + hr$ in $(\mathbf{Z}/q)[x]/P$.

e.g. typically $q = 2048$ in NTRU. Have natural ring maps from $(\mathbf{Z}/2048)[x]/(x^p - 1)$ to
$(\mathbf{Z}/2)[x]/(x^p - 1)$,
$(\mathbf{Z}/4)[x]/(x^p - 1)$,
$(\mathbf{Z}/8)[x]/(x^p - 1)$, etc.
Can attacker exploit these?

Maybe. Complicated. See 2004 Smart–Vercauteren–Silverman.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

nerally: Attacker applies
map $(\mathbf{Z}/q)[x]/P \to T$
quations $h = 3g/f$
$m + hr$ in $(\mathbf{Z}/q)[x]/P$.

cally $q = 2048$ in NTRU.
tural ring maps from
3)$[x]/(x^p - 1)$ to
]$/(x^p - 1)$,
]$/(x^p - 1)$,
]$/(x^p - 1)$, etc.
acker exploit these?

Complicated. See 2004
/ercauteren–Silverman.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LW
(2012 La
prove th
of the m
to the c
of LWE

ttacker applies
$q)[x]/P \to T$
$= 3g/f$
n $(\mathbf{Z}/q)[x]/P$.

2048 in NTRU.
maps from
$-\ 1)$ to
,
,
, etc.
oit these?

ted. See 2004
n–Silverman.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LWE religion
(2012 Langlois–St
prove that the arit
of the modulus q
to the computatio
of LWE and RLWE

plies

$\to T$

$/P$.

TRU.

n

004

an.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LWE religion, version 2
(2012 Langlois–Stehlé): "W
prove that the arithmetic fo
of the modulus q is irrelevar
to the computational hardne
of LWE and RLWE."

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

Ring-LWE religion, version 1: For "provable security", take prime q so that P splits completely in $\mathbf{Z}[x]/q$; i.e., have n different ring maps $(\mathbf{Z}/q)[x]/P \to \mathbf{Z}/q$.

Do these maps damage security? Fast attacks in some cases: 2014 Eisenträger–Hallgren–Lauter, 2015 Elias–Lauter–Ozman–Stange, 2016 Chen–Lauter–Stange.

Fast non-q-dependent attack by 2016 Castryck–Iliashenko–Vercauteren breaks 2015 ELOS cases but not 2016 CLS cases.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

But rounding adds noise, making attacks harder! The proof *limits* security gap but does not eliminate it.

/E religion, version 1: For
le security", take prime
t P splits completely in
i.e., have n different ring
$Z/q)[x]/P \to \mathbf{Z}/q$.

e maps damage security?
acks in some cases: 2014
ger–Hallgren–Lauter, 2015
uter–Ozman–Stange,
en–Lauter–Stange.

n-q-dependent attack
Castryck–Iliashenko–
eren breaks 2015 ELOS
t not 2016 CLS cases.

Ring-LWE religion, version 2
(2012 Langlois–Stehlé): "We
prove that the arithmetic form
of the modulus q is irrelevant
to the computational hardness
of LWE and RLWE."

Basic idea: "modulus switching"
from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker
multiplies by q'/q and rounds.

But rounding adds noise,
making attacks harder!
The proof *limits* security gap
but does not eliminate it.

We reco
that rem
i.e., cho

Field ($\mathbf{Z}$
to any s

, version 1: For
”, take prime
completely in
n different ring
$\rightarrow \mathbf{Z}/q$.

mage security?
me cases: 2014
ren–Lauter, 2015
an–Stange,
–Stange.

dent attack
-Iliashenko–
s 2015 ELOS
6 CLS cases.

Ring-LWE religion, version 2
(2012 Langlois–Stehlé): “We
prove that the arithmetic form
of the modulus q is irrelevant
to the computational hardness
of LWE and RLWE.”

Basic idea: “modulus switching”
from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker
multiplies by q'/q and rounds.

But rounding adds noise,
making attacks harder!
The proof *limits* security gap
but does not eliminate it.

We recommend: T
that remains irred
i.e., choose **inert r**

Field $(\mathbf{Z}/q)[x]/P$.
to any smaller nor

: For
me
y in
t ring

rity?
2014
, 2015
e,

k
–
OS
es.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

But rounding adds noise, making attacks harder!
The proof *limits* security gap but does not eliminate it.

We recommend: Take irred
that remains irred in $(\mathbf{Z}/q)[$
i.e., choose **inert modulus**

Field $(\mathbf{Z}/q)[x]/P$. No ring
to any smaller nonzero ring.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

But rounding adds noise, making attacks harder!
The proof *limits* security gap but does not eliminate it.

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

But rounding adds noise, making attacks harder!
The proof *limits* security gap but does not eliminate it.

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

Ring-LWE religion, version 2 (2012 Langlois–Stehlé): "We prove that the arithmetic form of the modulus q is irrelevant to the computational hardness of LWE and RLWE."

Basic idea: "modulus switching" from $\mathbf{Z}/q$ to $\mathbf{Z}/q'$. Attacker multiplies by q'/q and rounds.

But rounding adds noise, making attacks harder!
The proof *limits* security gap but does not eliminate it.

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

But we also recommend heresy: take P with **prime degree** p and with **large Galois group**, specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

/E religion, version 2
anglois–Stehlé): "We
at the arithmetic form
odulus q is irrelevant
omputational hardness
and RLWE."

ea: "modulus switching"
q to $\mathbf{Z}/q'$. Attacker
es by q'/q and rounds.

nding adds noise,
attacks harder!
of *limits* security gap
s not eliminate it.

We recommend: Take irred P
that remains irred in $(\mathbf{Z}/q)[x]$;
i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map
to any smaller nonzero ring.

So far this is compatible with
Ring-LWE religion, version 2.

But we also recommend heresy:
take P with **prime degree** p
and with **large Galois group**,
specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

2014.02,
To elimi
structure
of prime
subfield
polynom
very larg
the num
having a

, version 2
ehlé): "We
thmetic form
is irrelevant
nal hardness
E."

ulus switching"
. Attacker
and rounds.

s noise,
rder!
ecurity gap
nate it.

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

But we also recommend heresy: take P with **prime degree** p and with **large Galois group**, specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

2014.02, our 2nd a
To eliminate "wor
structures, use "a
of prime degree, s
subfield is $\mathbf{Q}$" and
polynomial $x^p - x$
very large Galois g
the number field is
having automorph

2
/e
rm
nt
ess

ning"
r
ds.

p

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

But we also recommend heresy: take P with **prime degree** p and with **large Galois group**, specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

2014.02, our 2nd announcen
To eliminate "worrisome"
structures, use "a number fi
of prime degree, so that the
subfield is $\mathbf{Q}$" and "an irredu
polynomial $x^p - x - 1$ with
very large Galois group, so t
the number field is very far
having automorphisms".

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

But we also recommend heresy: take P with **prime degree** p and with **large Galois group**, specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

We recommend: Take irred P that remains irred in $(\mathbf{Z}/q)[x]$; i.e., choose **inert modulus** q.

Field $(\mathbf{Z}/q)[x]/P$. No ring map to any smaller nonzero ring.

So far this is compatible with Ring-LWE religion, version 2.

But we also recommend heresy: take P with **prime degree** p and with **large Galois group**, specifically S_p, size $p!$.
Good example: $P = x^p - x - 1$.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

mmend: Take irred P
nains irred in $(\mathbf{Z}/q)[x]$;
ose **inert modulus** q.

$/q)[x]/P$. No ring map
maller nonzero ring.

his is compatible with
/E religion, version 2.

also recommend heresy:
with **prime degree** p
n **large Galois group**,
lly S_p, size $p!$.
ample: $P = x^p - x - 1$.

2014.02, our 2nd announcement:
To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10
Shepher
based sy
quantum

Take irred P
in $(\mathbf{Z}/q)[x]$;
modulus q.

No ring map
nzero ring.

patible with
, version 2.

mend heresy:
e **degree** p
lois group,
e $p!$.
$= x^p - x - 1$.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell-
Shepherd describe
based system "Sol
quantum poly-tim

P
$x]$;
q.
map
h
2.
esy:
p
p,
$- 1$.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is **Q**" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell–Groves–
Shepherd describe an ideal-l
based system "Soliloquy"; c
quantum poly-time key reco

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2014.02, our 2nd announcement: To eliminate “worrisome” structures, use “a number field of prime degree, so that the only subfield is $\mathbf{Q}$” and “an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms”.

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system “Soliloquy”; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2014.02, our 2nd announcement: To eliminate "worrisome" structures, use "a number field of prime degree, so that the only subfield is $\mathbf{Q}$" and "an irreducible polynomial $x^p - x - 1$ with a very large Galois group, so that the number field is very far from having automorphisms".

Subsequent attacks against several lattice-based systems have exploited these structures and have not been extended to our recommended rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

, our 2nd announcement:
nate "worrisome"
es, use "a number field
degree, so that the only
is $\mathbf{Q}$" and "an irreducible
ial $x^p - x - 1$ with a
ge Galois group, so that
ber field is very far from
utomorphisms".

ent attacks against
attice-based systems
ploited these structures
e not been extended
ecommended rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

SV/Solil
$k \geq 1$. D

Public k

Secret k
with $g\mathcal{R}$
i.e., shor
of the id

announcement:
risome"
number field
o that the only
"an irreducible
$- 1$ with a
group, so that
s very far from
isms".

ks against
ed systems
se structures
n extended
led rings.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

SV/Soliloquy para
$k \geq 1$. Define $\mathcal{R}$ =

Public key: prime

Secret key: short
with $g\mathcal{R} = q\mathcal{R} +$
i.e., short generato
of the ideal $q\mathcal{R} +$

ment:

eld

only

ucible

a

hat

from

s

res

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system “Soliloquy”; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

SV/Soliloquy parameter:

$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_2$

Public key: prime q and $c \in$

Secret key: short element g

with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;

i.e., short generator

of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

SV/Soliloquy parameter: $k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$ with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$; i.e., short generator of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

2014.10 Campbell–Groves–Shepherd describe an ideal-lattice-based system "Soliloquy"; claim quantum poly-time key recovery.

2010 Smart–Vercauteren system is practically identical to Soliloquy.

2009 Gentry system (simpler version described at STOC) has the same key-recovery problem.

2012 Garg–Gentry–Halevi multilinear maps have the same key-recovery problem (and many other security issues).

SV/Soliloquy parameter: $k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$ with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$; i.e., short generator of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to compute a generator of an ideal? See, e.g., 1993 Cohen textbook "A course in computational algebraic number theory".

Campbell–Groves–
d describe an ideal-lattice-
stem “Soliloquy”; claim
n poly-time key recovery.

nart–Vercauteren system is
lly identical to Soliloquy.

ntry system (simpler
described at STOC) has
e key-recovery problem.

rg–Gentry–Halevi
ear maps have the
y-recovery problem
ny other security issues).

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$
with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
i.e., short generator
of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to
compute a generator of an ideal?
See, e.g., 1993 Cohen textbook
“A course in computational
algebraic number theory”.

Smart–V
as takin

–Groves–
an ideal-lattice-
iloquy"; claim
e key recovery.

uteren system is
l to Soliloquy.

m (simpler
at STOC) has
very problem.

–Halevi
ave the
problem
security issues).

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$
with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
i.e., short generator
of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to
compute a generator of an ideal?
See, e.g., 1993 Cohen textbook
"A course in computational
algebraic number theory".

Smart–Vercautere
as taking exponen

attice-
laim
very.

stem is
oquy.

r
has
em.

sues).

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$
with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
i.e., short generator
of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to
compute a generator of an ideal?
See, e.g., 1993 Cohen textbook
"A course in computational
algebraic number theory".

Smart–Vercauteren dismiss t
as taking exponential time.

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$
with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
i.e., short generator
of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to
compute a generator of an ideal?
See, e.g., 1993 Cohen textbook
"A course in computational
algebraic number theory".

Smart–Vercauteren dismiss this
as taking exponential time.

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$
with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
i.e., short generator
of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to compute a generator of an ideal? See, e.g., 1993 Cohen textbook "A course in computational algebraic number theory".

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$ with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$; i.e., short generator of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to compute a generator of an ideal? See, e.g., 1993 Cohen textbook "A course in computational algebraic number theory".

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

SV/Soliloquy parameter:
$k \geq 1$. Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Public key: prime q and $c \in \mathbf{Z}/q$.

Secret key: short element $g \in \mathcal{R}$ with $g\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$; i.e., short generator of the ideal $q\mathcal{R} + (x - c)\mathcal{R}$.

But wait, isn't it known how to compute a generator of an ideal? See, e.g., 1993 Cohen textbook "A course in computational algebraic number theory".

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

oquy parameter:
Define $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

ey: prime q and $c \in \mathbf{Z}/q$.

ey: short element $g \in \mathcal{R}$
$\mathcal{R} = q\mathcal{R} + (x - c)\mathcal{R}$;
rt generator
deal $q\mathcal{R} + (x - c)\mathcal{R}$.

t, isn't it known how to
e a generator of an ideal?
., 1993 Cohen textbook
se in computational
c number theory".

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

Smart–V
this gene

Have ide
Want sh
Have g'
Know g'
But how

meter:

$= \mathbf{Z}[x]/\Phi_{2^k}$.

q and $c \in \mathbf{Z}/q$.

element $g \in \mathcal{R}$

$(x - c)\mathcal{R}$;

or

$(x - c)\mathcal{R}$.

known how to

tor of an ideal?

hen textbook

putational

theory".

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

Smart–Vercauteren

this generator as r

Have ideal I of $\mathcal{R}$

Want short g with

Have g' with $g'\mathcal{R}$

Know $g' = ug$ for

But how do we fin

k .

$\in \mathbf{Z}/q$.

$\in \mathcal{R}$

.

to

deal?

ook

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

Smart–Vercauteren also dism

this generator as not being s

Have ideal I of $\mathcal{R}$.

Want short g with $g\mathcal{R} = I$.

Have g' with $g'\mathcal{R} = I$.

Know $g' = ug$ for some $u \in$

But how do we find u?

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

Smart–Vercauteren dismiss this as taking exponential time.

It actually takes subexponential time. Same basic idea as NFS.

Campbell–Groves–Shepherd claim quantum poly time. Claim disputed by Biasse, not defended by CGS.

2016 Biasse–Song, building on 2014 Eisenträger–Hallgren–Kitaev–Song: different algorithm that takes quantum poly time.

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

$\operatorname{Log} g' = \operatorname{Log} u + \operatorname{Log} g$
where Log is Dirichlet's log map.

Dirichlet's unit theorem:
$\operatorname{Log} \mathcal{R}^*$ is a lattice, known dim.

Finding $\operatorname{Log} u$ is a closest-vector problem in this lattice.

Vercauteren dismiss this
g exponential time.

ly takes subexponential
ame basic idea as NFS.

ll–Groves–Shepherd
uantum poly time.
sputed by Biasse,
nded by CGS.

asse–Song, building on
senträger–Hallgren–
Song: different algorithm
es quantum poly time.

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

$\operatorname{Log} g' = \operatorname{Log} u + \operatorname{Log} g$
where Log is Dirichlet's log map.

Dirichlet's unit theorem:
$\operatorname{Log} \mathcal{R}^*$ is a lattice, known dim.

Finding $\operatorname{Log} u$ is a closest-vector problem in this lattice.

Campbe
“A simp
cyclotom
known.
$\mathcal{R}^*$] und
forms a
of this la
much bi
length o
g], so it
causally
any gen
e.g. via
algorithm

n dismiss this
tial time.

ubexponential
idea as NFS.

-Shepherd
ly time.
Biasse,
GS.

, building on
Hallgren–
erent algorithm
m poly time.

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

$\operatorname{Log} g' = \operatorname{Log} u + \operatorname{Log} g$
where Log is Dirichlet's log map.

Dirichlet's unit theorem:
$\operatorname{Log} \mathcal{R}^*$ is a lattice, known dim.

Finding $\operatorname{Log} u$ is a closest-vector problem in this lattice.

Campbell–Groves–
"A simple generat
cyclotomic units is
known. The image
$\mathcal{R}^*$] under the log
forms a lattice. Th
of this lattice turn
much bigger than
length of a private
g], so it is easy to
causally short priv
any generator of
e.g. via the LLL la
algorithm."

this

tial
FS.

on

rithm
ne.

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

$\operatorname{Log} g' = \operatorname{Log} u + \operatorname{Log} g$
where Log is Dirichlet's log map.

Dirichlet's unit theorem:
$\operatorname{Log} \mathcal{R}^*$ is a lattice, known dim.

Finding $\operatorname{Log} u$ is a closest-vector problem in this lattice.

Campbell–Groves–Shepherd:
"A simple generating set for
cyclotomic units is of course
known. The image of $\mathcal{O}^\times$ [i
$\mathcal{R}^*$] under the logarithm ma
forms a lattice. The determ
of this lattice turns out to b
much bigger than the typica
length of a private key α [i.e
g], so it is easy to recover th
causally short private key giv
any generator of $\alpha\mathcal{O}$ [i.e., I
e.g. via the LLL lattice redu
algorithm."

Smart–Vercauteren also dismiss this generator as not being *short*.

Have ideal I of $\mathcal{R}$.
Want short g with $g\mathcal{R} = I$.
Have g' with $g'\mathcal{R} = I$.
Know $g' = ug$ for some $u \in \mathcal{R}^*$.
But how do we find u?

$\operatorname{Log} g' = \operatorname{Log} u + \operatorname{Log} g$
where Log is Dirichlet's log map.

Dirichlet's unit theorem:
$\operatorname{Log} \mathcal{R}^*$ is a lattice, known dim.

Finding $\operatorname{Log} u$ is a closest-vector problem in this lattice.

Campbell–Groves–Shepherd:
"A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

Vercauteren also dismiss
erator as not being *short*.

eal I of $\mathcal{R}$.
ort g with $g\mathcal{R} = I$.
with $g'\mathcal{R} = I$.
$' = ug$ for some $u \in \mathcal{R}^*$.
do we find u?

$=$ Log u + Log g
og is Dirichlet's log map.

t's unit theorem:
is a lattice, known dim.

Log u is a closest-vector
in this lattice.

Campbell–Groves–Shepherd:
"A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$,
automor

Easy to

n also dismiss
not being *short*.

.

n $g\mathcal{R} = I$.

$= I$.

some $u \in \mathcal{R}^*$.

nd u?

Log g

hlet's log map.

eorem:

e, known dim.

closest-vector

ttice.

Campbell–Groves–Shepherd: "A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$, $x \mapsto x^5$,

automorphisms of

Easy to see $(1 - x$

miss

short.

$\mathcal{R}^*$.

map.

lim.

ector

Campbell–Groves–Shepherd:
"A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, et

automorphisms of $\mathcal{R} = \mathbf{Z}[x]$

Easy to see $(1-x^3)/(1-x)$

Campbell–Groves–Shepherd:
"A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

Campbell–Groves–Shepherd: "A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \left\{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\right\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Campbell–Groves–Shepherd: "A simple generating set for the cyclotomic units is of course known. The image of $\mathcal{O}^\times$ [i.e., $\mathcal{R}^*$] under the logarithm map forms a lattice. The determinant of this lattice turns out to be much bigger than the typical log-length of a private key α [i.e., g], so it is easy to recover the causally short private key given *any* generator of $\alpha\mathcal{O}$ [i.e., I], *e.g.* via the LLL lattice reduction algorithm."

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \left\{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\right\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Experiments confirm that SV is quickly broken by LLL using, e.g., 1997 Washington textbook basis for cyclotomic units.
Shortness of basis is critical; missing from bogus CGS analysis.

ll–Groves–Shepherd:
le generating set for the
nic units is of course
The image of $\mathcal{O}^\times$ [i.e.,
er the logarithm map
lattice. The determinant
attice turns out to be
gger than the typical log-
f a private key α [i.e.,
is easy to recover the
short private key given
erator of $\alpha\mathcal{O}$ [i.e., I],
the LLL lattice reduction
m."

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Experiments confirm that SV is quickly broken by LLL using, e.g., 1997 Washington textbook basis for cyclotomic units.
Shortness of basis is critical; missing from bogus CGS analysis.

Attacker
automor

2016 Alb
"A subfi
overstret
Cryptana
Graded
norms g
2016 Ch
main tec
is the re
a field to
traces g
an order

-Shepherd:
ing set for the
s of course
e of $\mathcal{O}^\times$ [i.e.,
arithm map
he determinant
s out to be
the typical log-
key α [i.e.,
recover the
ate key given
$\alpha\mathcal{O}$ [i.e., I],
ttice reduction

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Experiments confirm that SV is quickly broken by LLL using, e.g., 1997 Washington textbook basis for cyclotomic units.
Shortness of basis is critical; missing from bogus CGS analysis.

Attackers can also
automorphisms in

2016 Albrecht–Bai
"A subfield lattice
overstretched NTR
Cryptanalysis of so
Graded Encoding S
norms $g\sigma(g)$, and
2016 Cheon–Jeong
main technique of
is the reduction of
a field to one in a
traces $g + \sigma(g)$, w
an order-2 automo

the
e
.e.,
p
inant
e
l log-
e.,
he
ven
],
ction

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Experiments confirm that SV is quickly broken by LLL using, e.g., 1997 Washington textbook basis for cyclotomic units.
Shortness of basis is critical; missing from bogus CGS analysis.

Attackers can also use automorphisms in more ways

2016 Albrecht–Bai–Ducas
"A subfield lattice attack on
overstretched NTRU assump
Cryptanalysis of some FHE a
Graded Encoding Schemes"
norms $g\sigma(g)$, and independe
2016 Cheon–Jeong–Lee ("T
main technique of our algori
is the reduction of a problen
a field to one in a subfield")
traces $g + \sigma(g)$, where σ is
an order-2 automorphism.

$x \mapsto x^3$, $x \mapsto x^5$, $x \mapsto x^7$, etc. are automorphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

Easy to see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

"Cyclotomic units" are defined as $\mathcal{R}^* \cap \{\pm x^{e_0} \prod_i (1-x^i)^{e_i}\}$.
Weber's conjecture: all elements of $\mathcal{R}^*$ are cyclotomic units.

Experiments confirm that SV is quickly broken by LLL using, e.g., 1997 Washington textbook basis for cyclotomic units.
Shortness of basis is critical; missing from bogus CGS analysis.

Attackers can also use automorphisms in more ways.

2016 Albrecht–Bai–Ducas "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes" use norms $g\sigma(g)$, and independently 2016 Cheon–Jeong–Lee ("The main technique of our algorithm is the reduction of a problem on a field to one in a subfield") use traces $g + \sigma(g)$, where σ is an order-2 automorphism.

$x \mapsto x^5$, $x \mapsto x^7$, etc. are
rphisms of $\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

see $(1-x^3)/(1-x) \in \mathcal{R}^*$.

omic units" are defined as
$\pm x^{e_0} \prod_i (1-x^i)^{e_i}\}$.
conjecture: all elements
re cyclotomic units.

ents confirm that SV is
broken by LLL using, e.g.,
ashington textbook
cyclotomic units.
ss of basis is critical;
from bogus CGS analysis.

Attackers can also use automorphisms in more ways.

2016 Albrecht–Bai–Ducas "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes" use norms $g\sigma(g)$, and independently 2016 Cheon–Jeong–Lee ("The main technique of our algorithm is the reduction of a problem on a field to one in a subfield") use traces $g + \sigma(g)$, where σ is an order-2 automorphism.

We reco
the choi
ideal-lat

Requirin
minimize

Requirin
S_p maxi
automor
the smal
roots of

All avail
this resc
and neve

$x \mapsto x^7$, etc. are

$\mathcal{R} = \mathbf{Z}[x]/\Phi_{2^k}$.

${}^3)/(1-x) \in \mathcal{R}^*$.

" are defined as

$- x^i)^{e_i}\}$.

e: all elements

mic units.

rm that SV is

LLL using, e.g.,

textbook

ic units.

is critical;

us CGS analysis.

Attackers can also use automorphisms in more ways.

2016 Albrecht–Bai–Ducas "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes" use norms $g\sigma(g)$, and independently 2016 Cheon–Jeong–Lee ("The main technique of our algorithm is the reduction of a problem on a field to one in a subfield") use traces $g + \sigma(g)$, where σ is an order-2 automorphism.

We recommend ch

the choice of rings

ideal-lattice-based

Requiring prime de

minimizes number

Requiring Galois g

S_p maximizes diffi

automorphism con

the smallest field c

roots of P has deg

All available evide

this rescues *some*

and never hurts se

tc. are

$/\Phi_{2^k}$.

$\in \mathcal{R}^*$.

ned as

nents

V is

, e.g.,

alysis.

Attackers can also use automorphisms in more ways.

2016 Albrecht–Bai–Ducas "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes" use norms $g\sigma(g)$, and independently 2016 Cheon–Jeong–Lee ("The main technique of our algorithm is the reduction of a problem on a field to one in a subfield") use traces $g + \sigma(g)$, where σ is an order-2 automorphism.

We recommend changing the choice of rings in ideal-lattice-based cryptogra

Requiring prime degree p minimizes number of subfiel

Requiring Galois group S_p maximizes difficulty of automorphism computations the smallest field containing roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

Attackers can also use automorphisms in more ways.

2016 Albrecht–Bai–Ducas "A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and Graded Encoding Schemes" use norms $g\sigma(g)$, and independently 2016 Cheon–Jeong–Lee ("The main technique of our algorithm is the reduction of a problem on a field to one in a subfield") use traces $g + \sigma(g)$, where σ is an order-2 automorphism.

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

rs can also use
rphisms in more ways.

brecht–Bai–Ducas
eld lattice attack on
tched NTRU assumptions:
alysis of some FHE and
Encoding Schemes" use
$\sigma(g)$, and independently
eon–Jeong–Lee ("The
chnique of our algorithm
duction of a problem on
o one in a subfield") use
$+ \sigma(g)$, where σ is
-2 automorphism.

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The imp

"If you'r
structure
visible p
Use LWE

use
more ways.

i–Ducas
attack on
RU assumptions:
ome FHE and
Schemes” use
independently
g–Lee (“The
our algorithm
f a problem on
subfield”) use
where σ is
orphism.

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The importance of

“If you’re so worri
structure, why are
visible polynomial
Use LWE, or class

s.

n
ptions:
and
use
ently
he
thm
n on
use

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The importance of efficiency

“If you’re so worried about structure, why are you tolera visible polynomial structure? Use LWE, or classic McEliec

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes— but huge costs in network traffic. Is this affordable?

We recommend changing the choice of rings in ideal-lattice-based cryptography.

Requiring prime degree p minimizes number of subfields.

Requiring Galois group S_p maximizes difficulty of automorphism computations: e.g., the smallest field containing all roots of P has degree $p!$.

All available evidence is that this rescues *some* systems and never hurts security.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes—but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

mmend changing
ce of rings in
tice-based cryptography.

g prime degree p
es number of subfields.

g Galois group
mizes difficulty of
rphism computations: e.g.,
llest field containing all
P has degree $p!$.

able evidence is that
ues *some* systems
er hurts security.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes—but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

Convent
Rings (**Z**
with q m
extremel
NTRU P
several t
Is this af

anging
s in
cryptography.

egree p
of subfields.

roup
culty of
mputations: e.g.,
containing all
gree $p!$.

nce is that
systems
curity.

The importance of efficiency

“If you’re so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!”

Maybe better security, yes— but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

Conventional wisd
Rings $(\mathbf{Z}/q)[x]/\Phi$
with q mod 2^{k+1}
extremely fast FF
NTRU Prime ring
several times slow
Is this affordable?

phy.

ds.

s: e.g.,

all

t

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes— but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

Conventional wisdom:
Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$
with $q \bmod 2^{k+1} = 1$ allow
extremely fast FFT-based m
NTRU Prime rings will be
several times slower.
Is this affordable? etc.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes—but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

Conventional wisdom: Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$ with $q \bmod 2^{k+1} = 1$ allow extremely fast FFT-based mults. NTRU Prime rings will be several times slower. Is this affordable? etc.

The importance of efficiency

"If you're so worried about structure, why are you tolerating visible polynomial structure? Use LWE, or classic McEliece!"

Maybe better security, yes— but huge costs in network traffic. Is this affordable?

If it is, would we gain more security from larger polynomials? Larger impact on known attacks, maybe also on unknown attacks. Not clear what to recommend.

Conventional wisdom: Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$ with $q \bmod 2^{k+1} = 1$ allow extremely fast FFT-based mults. NTRU Prime rings will be several times slower. Is this affordable? etc.

But we have shown that an optimized combination of Karatsuba and Toom is also extremely fast at crypto sizes. Hard to find any applications that will notice the differences. And we *improve* network traffic.

ortance of efficiency

re so worried about
e, why are you tolerating
olynomial structure?
E, or classic McEliece!"

better security, yes—
e costs in network traffic.
ffordable?

would we gain more
from larger polynomials?
mpact on known attacks,
lso on unknown attacks.
r what to recommend.

Conventional wisdom:
Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$
with $q \bmod 2^{k+1} = 1$ allow
extremely fast FFT-based mults.
NTRU Prime rings will be
several times slower.
Is this affordable? etc.

But we have shown that
an optimized combination of
Karatsuba and Toom is also
extremely fast at crypto sizes.
Hard to find any applications
that will notice the differences.
And we *improve* network traffic.

What yo

Streamli
an optim

The desi
lattice-b

Security
Prime: r
attacks,

Paramet

Public-k
unauthe

And mor

f efficiency

ed about
you tolerating
structure?
ic McEliece!"

rity, yes—
network traffic.

gain more
er polynomials?
known attacks,
known attacks.
recommend.

Conventional wisdom:
Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$
with $q \bmod 2^{k+1} = 1$ allow
extremely fast FFT-based mults.
NTRU Prime rings will be
several times slower.
Is this affordable? etc.

But we have shown that
an optimized combination of
Karatsuba and Toom is also
extremely fast at crypto sizes.
Hard to find any applications
that will notice the differences.
And we *improve* network traffic.

What you find in p

Streamlined NTRU
an optimized crypt

The design space o
lattice-based encry

Security of Stream
Prime: meet-in-th
attacks, lattice att

Parameters.

Public-key encrypt
unauthenticated ke

And more!

y

ating
,
e!”

-
raffic.

nials?
acks,
acks.
nd.

Conventional wisdom:
Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$
with $q \bmod 2^{k+1} = 1$ allow
extremely fast FFT-based mults.
NTRU Prime rings will be
several times slower.
Is this affordable? etc.

But we have shown that
an optimized combination of
Karatsuba and Toom is also
extremely fast at crypto sizes.
Hard to find any applications
that will notice the differences.
And we *improve* network traffic.

What you find in paper

Streamlined NTRU Prime:
an optimized cryptosystem.

The design space of
lattice-based encryption.

Security of Streamlined NTR
Prime: meet-in-the-middle
attacks, lattice attacks, etc.

Parameters.

Public-key encryption vs.
unauthenticated key exchan

And more!

Conventional wisdom:
Rings $(\mathbf{Z}/q)[x]/\Phi_{2^k}$
with $q \bmod 2^{k+1} = 1$ allow
extremely fast FFT-based mults.
NTRU Prime rings will be
several times slower.
Is this affordable? etc.

But we have shown that
an optimized combination of
Karatsuba and Toom is also
extremely fast at crypto sizes.
Hard to find any applications
that will notice the differences.
And we *improve* network traffic.

What you find in paper

Streamlined NTRU Prime:
an optimized cryptosystem.

The design space of
lattice-based encryption.

Security of Streamlined NTRU
Prime: meet-in-the-middle
attacks, lattice attacks, etc.

Parameters.

Public-key encryption vs.
unauthenticated key exchange.

And more!