Is $2^{255} - 19$ big enough?

Generate public keys
on a "strong" elliptic curve E
over the field $\mathbf{Z}/(2^{255} - 19)$.
Is that safe?

"Size does matter!"

What marketing says

56-bit crypto: Broken.
128-bit crypto: Okay.
256-bit crypto: High security!
512-bit crypto: Broken.
1024-bit crypto: Shaky.

$2^{255} - 19$ must be, um, 256 bits.
Fantastic!
Best possible security level.

hough?

eys

otic curve E

$2^{255} - 19)$.

!"

What marketing says

56-bit crypto: Broken.

128-bit crypto: Okay.

256-bit crypto: High security!

512-bit crypto: Broken.

1024-bit crypto: Shaky.

$2^{255} - 19$ must be, um, 256 bits.

Fantastic!

Best possible security level.

What NSA says

NSA approves pro
"classified or missi
national security in

NSA wants "ellipt
GF(p), where p is
greater than 2^{255}."

So $2^{255} + 95$ is fin
for national securit
but $2^{255} - 19$ is n

What marketing says

56-bit crypto: Broken.

128-bit crypto: Okay.

256-bit crypto: High security!

512-bit crypto: Broken.

1024-bit crypto: Shaky.

$2^{255} - 19$ must be, um, 256 bits.

Fantastic!

Best possible security level.

What NSA says

NSA approves products for "classified or mission critical national security information."

NSA wants "elliptic curves over $\mathrm{GF}(p)$, where p is a prime number greater than 2^{255}."

So $2^{255} + 95$ is fine for national security information but $2^{255} - 19$ is not.

ays

oken.

kay.

gh security!

roken.

Shaky.

e, um, 256 bits.

rity level.

What NSA says

NSA approves products for "classified or mission critical national security information."

NSA wants "elliptic curves over GF(p), where p is a prime number greater than 2^{255}."

So $2^{255}+95$ is fine for national security information but $2^{255}-19$ is not.

What NIST says

128-bit AES keys

ECC primes with "

the amount of wor

"break the algorit

is approximately th

namely 2^{128} opera

by best techniques

What NSA says

NSA approves products for "classified or mission critical national security information."

NSA wants "elliptic curves over GF(p), where p is a prime number greater than 2^{255}."

So $2^{255} + 95$ is fine
for national security information
but $2^{255} - 19$ is not.

What NIST says

128-bit AES keys "correspond" to ECC primes with "256-383" bits:
the amount of work needed to "break the algorithms"
is approximately the same,
namely 2^{128} operations,
by best techniques known.

ducts for
on critical
nformation.”

ic curves over
a prime number
,

e
ty information
ot.

What NIST says

128-bit AES keys “correspond” to ECC primes with “256-383” bits: the amount of work needed to “break the algorithms” is approximately the same, namely 2^{128} operations, by best techniques known.

What I say

Given $H(k) =$ AE
using $\approx 2^{127}$ AES

Given $H(k_1), H(k_2$
find *all* k_i using a
AES evaluations.

Or find *some* k_i u
evaluations.

Standard algorithm
negligible commun
perfect parallelizat
`cr.yp.to/papers`
`#bruteforce`

What NIST says

128-bit AES keys "correspond" to ECC primes with "256-383" bits: the amount of work needed to "break the algorithms" is approximately the same, namely 2^{128} operations, by best techniques known.

What I say

Given $H(k) = \text{AES}_k(0)$, find k using $\approx 2^{127}$ AES evaluations.

Given $H(k_1), H(k_2), \ldots, H(k_{2^{40}})$, find *all* k_i using a *total* of $\approx 2^{127}$ AES evaluations.

Or find *some* k_i using $\approx 2^{87}$ AES evaluations.

Standard algorithms have negligible communication and perfect parallelization: see, e.g., `cr.yp.to/papers.html#bruteforce`

"correspond" to
"256-383" bits:
rk needed to
hms"
he same,
tions,
s known.

What I say

Given $H(k) = \text{AES}_k(0)$, find k using $\approx 2^{127}$ AES evaluations.

Given $H(k_1), H(k_2), \ldots, H(k_{2^{40}})$, find *all* k_i using a *total* of $\approx 2^{127}$ AES evaluations.

Or find *some* k_i using $\approx 2^{87}$ AES evaluations.

Standard algorithms have negligible communication and perfect parallelization: see, e.g., `cr.yp.to/papers.html#bruteforce`

Given public key o
255-bit elliptic cur
find secret key
using $\approx 2^{127}$ addit

Given 2^{40} public k
find all secret keys
using $\approx 2^{147}$ addit

Finding *some* key
as finding first key
$\approx 2^{127}$ additions.
by random self-red

See, e.g., Kuhn an

What I say

Given $H(k) = \text{AES}_k(0)$, find k using $\approx 2^{127}$ AES evaluations.

Given $H(k_1), H(k_2), \ldots, H(k_{2^{40}})$, find *all* k_i using a *total* of $\approx 2^{127}$ AES evaluations.

Or find *some* k_i using $\approx 2^{87}$ AES evaluations.

Standard algorithms have negligible communication and perfect parallelization: see, e.g., `cr.yp.to/papers.html#bruteforce`

Given public key on 255-bit elliptic curve E, find secret key using $\approx 2^{127}$ additions on E.

Given 2^{40} public keys, find all secret keys using $\approx 2^{147}$ additions on E.

Finding *some* key is as hard as finding first key: $\approx 2^{127}$ additions. Easily prove by random self-reduction.

See, e.g., Kuhn and Struik, 2001.

ES$_k(0)$, find k
evaluations.

$_2), \ldots, H(k_{2^{40}})$,
total of $\approx 2^{127}$

sing $\approx 2^{87}$ AES

ns have
nication and
tion: see, e.g.,
`s.html`

Given public key on
255-bit elliptic curve E,
find secret key
using $\approx 2^{127}$ additions on E.

Given 2^{40} public keys,
find all secret keys
using $\approx 2^{147}$ additions on E.

Finding *some* key is as hard
as finding first key:
$\approx 2^{127}$ additions. Easily prove
by random self-reduction.

See, e.g., Kuhn and Struik, 2001.

Even worse for AE
can try much less
Success chance dr

For elliptic curves,
drops quadratically

Bottom line: 128-
not comparable in
to 255-bit elliptic-

Is $2^{255} - 19$ big en
Is 128-bit AES saf

Given public key on
255-bit elliptic curve E,
find secret key
using $\approx 2^{127}$ additions on E.

Given 2^{40} public keys,
find all secret keys
using $\approx 2^{147}$ additions on E.

Finding *some* key is as hard
as finding first key:
$\approx 2^{127}$ additions. Easily prove
by random self-reduction.

See, e.g., Kuhn and Struik, 2001.

Even worse for AES: Attacker
can try much less computation.
Success chance drops linearly.

For elliptic curves, success chance
drops quadratically.

Bottom line: 128-bit AES keys are
not comparable in security
to 255-bit elliptic-curve keys.

Is $2^{255} - 19$ big enough? Yes.
Is 128-bit AES safe? Unclear.