Security dangers of the NIST curves

D. J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Joint work with:
Tanja Lange
Technische Universiteit Eindhoven

The NIST curves were designed to make DLP difficult.

Or were they?

“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”
Security dangers of the NIST curves

D. J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Joint work with:
Tanja Lange
Technische Universiteit Eindhoven

The NIST curves were designed to make DLP difficult.

Or were they?

“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”
The NIST curves were designed to make DLP difficult. Or were they?

“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.
The NIST curves were designed to make DLP difficult. Or were they?

“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.
Security dangers of the NIST curves

D. J. Bernstein
University of Illinois at Chicago &
Technische Universiteit Eindhoven

Joint work with:
Tanja Lange
Technische Universiteit Eindhoven

The NIST curves were designed
to make DLP difficult.
Or were they?

“ECC Brainpool Standard Curves
and Curve Generation version
1.0”, 2005.10.19: “The choice
of the seeds from which the curve
parameters have been derived is
not motivated leaving an essential
part of the security analysis open.”

Bruce Schneier, “NSA
surveillance: A guide to
staying secure”, The Guardian,
2013.09.06: “Prefer conventional
discrete-log-based systems over
elliptic-curve systems; the latter
have constants that the NSA
influences when they can.”

But that’s not our main point.
As far as we know today,
NIST-curve DLP is secure.
“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.
“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.

Here’s our main point: NIST-curve ECC is much less secure than NIST-curve DLP.
But that’s not our main point. As far as we know today, NIST-curve DLP is secure.

Here’s our main point: **NIST-curve ECC is much less secure than NIST-curve DLP.**

If you use the NIST curves, you’re probably doing it wrong.

Your code produces incorrect results for some rare curve points; leaks secret data when the input isn’t a curve point; leaks secret data through cache timing; etc.

“ECC Brainpool Standard Curves and Curve Generation version 1.0”, 2005.10.19: “The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.”

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”
The choice of the seeds from which the curve parameters have been derived is motivated leaving an essential part of the security analysis open.

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.

Here’s our main point: **NIST-curve ECC is much less secure than NIST-curve DLP.**

If you use the NIST curves, you’re probably doing it wrong.

Your code produces incorrect results for some rare curve points; leaks secret data when the input isn’t a curve point; leaks secret data through cache timing; etc.

These problems are exploitable by attackers.

These attacks are against real protocols; DLP is non-interactive; computes \( nP \) correctly; reveals only \( nP \).

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.
standard Curves
and Curve Generation version
1.0", 2005.10.19: “The choice
of the seeds from which the curve
parameters have been derived is
leaving an essential
part of the security analysis open.”

Bruce Schneier, “NSA
surveillance: A guide to
staying secure”, The Guardian,
2013.09.06: “Prefer conventional
discrete-log-based systems over
elliptic-curve systems; the latter
have constants that the NSA
influences when they can.”

But that’s not our main point.
As far as we know today,
NIST-curve DLP is secure.

Here’s our main point:
NIST-curve ECC is much less
secure than NIST-curve DLP.

If you use the NIST curves,
you’re probably doing it wrong.

Your code produces incorrect
results for some rare curve points;
leaks secret data when the input
isn’t a curve point; leaks secret
data through cache timing; etc.

These problems are
exploitable by attackers.

These attacks are against
real protocols, not DLP.
DLP is non-interactive;
computes $nP$ correctly;
reveals only $nP$.

Real protocols
handle attacker-controlled input;
have failure cases;
reveal timing.

Attacker exploits these
gaps.
The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.

Bruce Schneier, “NSA surveillance: A guide to staying secure”, The Guardian, 2013.09.06: “Prefer conventional discrete-log-based systems over elliptic-curve systems; the latter have constants that the NSA influences when they can.”

But that’s not our main point. As far as we know today, NIST-curve DLP is secure.

Here’s our main point: **NIST-curve ECC is much less secure than NIST-curve DLP.**

If you use the NIST curves, you’re probably doing it wrong.

Your code produces incorrect results for some rare curve points; leaks secret data when the input isn’t a curve point; leaks secret data through cache timing; etc.

These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP. DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.
But that’s not our main point. As far as we know today, NIST-curve DLP is secure.

Here’s our main point: **NIST-curve ECC is much less secure than NIST-curve DLP.**

If you use the NIST curves, you’re probably doing it wrong.

Your code produces incorrect results for some rare curve points; leaks secret data when the input isn’t a curve point; leaks secret data through cache timing; etc.

These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.
But that's not our main point. As far as we know today, NIST-curve DLP is secure. Here's our main point: curve ECC is much less secure than NIST-curve DLP. If you use the NIST curves, you're probably doing it wrong. Your code produces incorrect results for some rare curve points; leaks secret data when the input isn't a curve point; leaks secret data through cache timing; etc.

These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP. DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing. Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.
But that's not our main point. As far as we know today, NIST-curve DLP is secure. Here's our main point: NIST-curve ECC is much less secure than NIST-curve DLP. If you use the NIST curves, you're probably doing it wrong.

These problems are exploitable by attackers.
These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.
Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.
But that's not our main point. As far as we know today, NIST-curve DLP is secure. Here's our main point: NIST-curve ECC is much less secure than NIST-curve DLP. If you use the NIST curves, you're probably doing it wrong. Your code produces incorrect results for some rare curve points; leaks secret data when the input is not a curve point; leaks secret data through cache timing; etc. These problems are exploitable by attackers. These attacks are against real protocols, not against DLP. DLP is non-interactive; computes $nP$ correctly; reveals only $nP$. Real protocols handle attacker-controlled input; have failure cases; reveal timing. Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.
These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.
These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.

Sensible security engineering: Design curves for ECC security, not just for DLP security.
These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes $nP$ correctly; reveals only $nP$.

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.

Sensible security engineering: Design curves for ECC security, not just for DLP security.

Detailed analysis online now (+ white paper coming soon): cr.yp.to/talks/2013.05.31/slides-dan+tanja-20130531-4x3.pdf
These problems are exploitable by attackers. These attacks are against real protocols, not against DLP. DLP is non-interactive; computes $nP$ correctly; reveals only $nP$. Real protocols handle attacker-controlled input; have failure cases; reveal timing. Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.

Sensible security engineering: **Design curves for ECC security, not just for DLP security.**

Detailed analysis online now (+ white paper coming soon): cr.yp.to/talks/2013.05.31/slides-dan+tanja-20130531-4x3.pdf

⇒ Use Curve25519.
These problems are exploitable by attackers.

These attacks are against real protocols, not against DLP.

DLP is non-interactive; computes \( nP \) correctly; reveals only \( nP \).

Real protocols handle attacker-controlled input; have failure cases; reveal timing.

Attacker exploits these gaps.

Can NIST-curve ECC be safe? Theoretically, but hard to do; highly fragile; unintelligent use of limited security resources.

Sensible security engineering: Design curves for ECC security, not just for DLP security.

Detailed analysis online now (+ white paper coming soon):
cr.yp.to/talks/2013.05.31/slides-dan+tanja-20130531-4x3.pdf

⇒ Use Curve25519. Or \( x^2 + y^2 = 1 + 3617x^2y^2 \mod 2^{414} - 17 \).