(Picture credit: Reuters.)

How to manipulate standards

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

Chinese government
is under attack from
terrorists in Hong Kong.

(Picture credit: Reuters.)

How to manipulate standards

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

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

credit: Reuters.)

---

manipulate standards

. Bernstein
ty of Illinois at Chicago &
che Universiteit Eindhoven

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept
(Also ex

How doe
relate to

euters.)

---

e standards

n

is at Chicago &

siteit Eindhoven

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept the ciphe
(Also exploit meta

How does the ciph
relate to the plain

ls

ago &
hoven

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Chinese government
is under attack from
terrorists in Hong Kong.

Goal of this talk:
Help the government
decrypt the terrorists'
encrypted communications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

government
attack from
s in Hong Kong.

this talk:
e government
the terrorists'
ed communications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are ther
to find p
given AE

Extensiv
in public
Doesn't

nt
om
Kong.

ent
sts'
nications.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are there better w
to find plaintext
given AES ciphert

Extensively studied
in public literature
Doesn't look good

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Intercept the ciphertext.
(Also exploit metadata etc.)

How does the ciphertext
relate to the plaintext?

Maybe 56-bit DES.
Feasible to search
all 2^{56} possible keys,
check plaintext plausibility.

Maybe 128-bit AES.
Feasible search is unlikely
to find this target's key.
(But can improve probability
by batching many targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

t the ciphertext.
ploit metadata etc.)

es the ciphertext
the plaintext?

56-bit DES.
to search
ossible keys,
aintext plausibility.

128-bit AES.
search is unlikely
his target's key.
n improve probability
ning many targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

Standard
terrorists
we see c

Maybe t
compute
Unintent
With ou

ertext.
ndata etc.)

nertext
text?
s.

ys,
ausibility.

tS.
unlikely
s key.
probability
targets.)

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

Standard security
terrorists compute
we see cipher outp

Maybe terrorists o
compute somethin
Unintentionally: "
With our help: "fa

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

Standard security model say
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionall
compute something differen
Unintentionally: "bugs".
With our help: "faults".

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Are there better ways
to find plaintext
given AES ciphertext?

Extensively studied
in public literature.
Doesn't look good for us.

Maybe we're smarter and
can find something better
than what's in the literature.

Maybe there are other
parts of the system
that have been less studied,
are easier for us to break.

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Maybe we actually see
more than cipher output.
"Side channels": e.g.,
plaintext or key is visible
through power consumption
or electromagnetic radiation.

e better ways
plaintext
ES ciphertext?

ely studied
c literature.
look good for us.

we're smarter and
something better
at's in the literature.

here are other
the system
e been less studied,
er for us to break.

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Maybe we actually see
more than cipher output.
"Side channels": e.g.,
plaintext or key is visible
through power consumption
or electromagnetic radiation.

How do
agree up

Maybe *s*
Terrorist
produce
a randor

Maybe w
See Tan

Maybe t
stored o
and we
Lack of
(aka "fo

ways

ext?

d

.

l for us.

ter and

g better

literature.

ther

n

ss studied,

o break.

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Maybe we actually see
more than cipher output.
"Side channels": e.g.,
plaintext or key is visible
through power consumption
or electromagnetic radiation.

How do the terror
agree upon an AE

Maybe *secret-key*
Terrorists Alice an
produce 128-bit ke
a random-number

Maybe we can bre
See Tanja Lange's

Maybe the key is s
stored on Bob's co
and we can grab c
Lack of "key erasu
(aka "forward secr

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Maybe we actually see
more than cipher output.
"Side channels": e.g.,
plaintext or key is visible
through power consumption
or electromagnetic radiation.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptograp*
Terrorists Alice and Bob me
produce 128-bit key using
a random-number generator

Maybe we can break this RN
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

Standard security model says:
terrorists compute cipher;
we see cipher output.

Maybe terrorists occasionally
compute something different.
Unintentionally: "bugs".
With our help: "faults".

Maybe we actually see
more than cipher output.
"Side channels": e.g.,
plaintext or key is visible
through power consumption
or electromagnetic radiation.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptography*.
Terrorists Alice and Bob meet,
produce 128-bit key using
a random-number generator.

Maybe we can break this RNG.
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

d security model says:
s compute cipher;
ipher output.

errorists occasionally
e something different.
tionally: "bugs".
r help: "faults".

we actually see
an cipher output.
annels": e.g.,
t or key is visible
power consumption
omagnetic radiation.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptography*.
Terrorists Alice and Bob meet,
produce 128-bit key using
a random-number generator.

Maybe we can break this RNG.
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

Maybe *p*

e.g. ECD
using sta
on an el

1. Alice
sends *aP*
2. Bob
sends *bP*
3. Alice
4. Bob
5. Alice
abP into

model says:
cipher;
out.

ccasionally
g different.
bugs".
aults".

y see
output.
e.g.,
visible
nsumption
c radiation.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptography*.
Terrorists Alice and Bob meet,
produce 128-bit key using
a random-number generator.

Maybe we can break this RNG.
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

Maybe *public-key*

e.g. $\mathrm{ECDH}_{E,P}$ key
using standard poi
on an elliptic curv

1. Alice generates
sends aP on E.
2. Bob generates
sends bP on E.
3. Alice computes
4. Bob computes
5. Alice and Bob
abP into AES key

s:

y

t.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptography*.
Terrorists Alice and Bob meet,
produce 128-bit key using
a random-number generator.

Maybe we can break this RNG.
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

Maybe *public-key cryptogra*

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

How do the terrorists
agree upon an AES key?

Maybe *secret-key cryptography*.
Terrorists Alice and Bob meet,
produce 128-bit key using
a random-number generator.

Maybe we can break this RNG.
See Tanja Lange's talk.

Maybe the key is still
stored on Bob's computer
and we can grab computer.
Lack of "key erasure"
(aka "forward secrecy").

Maybe *public-key cryptography*.

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

the terrorists
on an AES key?

secret-key cryptography.
ts Alice and Bob meet,
128-bit key using
m-number generator.

we can break this RNG.
ja Lange's talk.

he key is still
n Bob's computer
can grab computer.
"key erasure"
rward secrecy").

Maybe *public-key cryptography*.

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

Maybe w

Maybe w
Hard if A

Maybe w
Hard if E
(Not co
Alice, Bo

Maybe w
compute

Maybe w
compute

ists
S key?

cryptography.
d Bob meet,
ey using
generator.

ak this RNG.
talk.

still
omputer
computer.
ure"
recy").

Maybe *public-key cryptography*.

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

Maybe we can bre

Maybe we can gra
Hard if Alice disca

Maybe we can *mo*
Hard if Bob alread
(Not compatible w
Alice, Bob use two

Maybe we can "br
compute a from a

Maybe we can "br
compute abP from

phy.
et,
.
NG.

Maybe *public-key cryptography*.

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

Maybe we can break RNG fo

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows i
(Not compatible with discar
Alice, Bob use two DH layer

Maybe we can "break ECDL
compute a from aP.

Maybe we can "break ECDH
compute abP from aP, bP.

Maybe *public-key cryptography*.

e.g. $\mathrm{ECDH}_{E,P}$ key exchange
using standard point P
on an elliptic curve E:

1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ublic-key cryptography.

$DH_{E,P}$ key exchange
andard point P
liptic curve E:

generates secret a,
P on E.
generates secret b,
P on E.
computes abP.
computes abP.
and Bob convert
AES key.

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/E
depends

How did
which cu

cryptography.

exchange
int P
e E:

secret a,

secret b,

abP.
abP.
convert
.

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH diffi
depends on curve

How did terrorists
which curve E to

phy.

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Maybe we can break RNG for a.

Maybe we can grab a.
Hard if Alice discarded it.

Maybe we can *modify* aP.
Hard if Bob already knows it.
(Not compatible with discard $\Rightarrow$
Alice, Bob use two DH layers.)

Maybe we can "break ECDL":
compute a from aP.

Maybe we can "break ECDH":
compute abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Did they screw up? (See TLS.)
Can we influence this?

we can break RNG for a.

we can grab a.
Alice discarded it.

we can *modify* aP.
Bob already knows it.
mpatible with discard $\Rightarrow$
ob use two DH layers.)

we can "break ECDL":
e a from aP.

we can "break ECDH":
e abP from aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Did they screw up? (See TLS.)
Can we influence this?

Move to
model o
e.g. prot

-1. Jer
0. Publi
1. Alice
sends aP
2. Bob
sends bP
3. Alice
4. Bob
5. Alice
abP into

ak RNG for a.

b a.

rded it.

odify aP.

ly knows it.

with discard $\Rightarrow$
o DH layers.)

reak ECDL":
P.

reak ECDH":
n aP, bP.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Did they screw up? (See TLS.)
Can we influence this?

Move towards mor
model of cryptogra
e.g. protocol ECD

-1. Jerry generat
0. Public checks W
1. Alice generates
sends aP on E.
2. Bob generates
sends bP on E.
3. Alice computes
4. Bob computes
5. Alice and Bob
abP into AES key

or a.

t.

d $\Rightarrow$

rs.)

”:

H”:

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Did they screw up? (See TLS.)
Can we influence this?

Move towards more accurate
model of cryptography.
e.g. protocol ECDH$_V$:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S)$.
1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

ECDL/ECDH difficulty
depends on curve E.

How did terrorists decide
which curve E to use?

How did terrorists decide
to use ECDH instead of
another public-key protocol?

How did terrorists decide
to use AES instead of
another secret-key cipher?

Did they screw up? (See TLS.)
Can we influence this?

Move towards more accurate
model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

ECDH difficulty
on curve E.

terrorists decide
urve E to use?

terrorists decide
CDH instead of
public-key protocol?

terrorists decide
AES instead of
secret-key cipher?

y screw up? (See TLS.)
influence this?

Move towards more accurate
model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

What is
Which c

culty
E.

decide
use?

decide
ead of
protocol?

decide
d of
cipher?

? (See TLS.)
e this?

Move towards more accurate
model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a,
sends aP on E.
2. Bob generates secret b,
sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert
abP into AES key.

What is V?
Which curves will

Move towards more accurate model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a, sends aP on E.
2. Bob generates secret b, sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert abP into AES key.

What is V?
Which curves will public acc

S.)

Move towards more accurate model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a, sends aP on E.
2. Bob generates secret b, sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert abP into AES key.

What is V?
Which curves will public accept?

Move towards more accurate model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a, sends aP on E.
2. Bob generates secret b, sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert abP into AES key.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

Move towards more accurate model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a, sends aP on E.
2. Bob generates secret b, sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert abP into AES key.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Move towards more accurate model of cryptography.
e.g. protocol ECDH_V:

-1. Jerry generates E, P, S.
0. Public checks $V(E, P, S) = 1$.
1. Alice generates secret a, sends aP on E.
2. Bob generates secret b, sends bP on E.
3. Alice computes abP.
4. Bob computes abP.
5. Alice and Bob convert abP into AES key.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature fails to formalize any of this.
Also fails to formalize analogous questions about selecting ciphers, protocols, etc.

wards more accurate
f cryptography.
tocol ECDH$_V$:

ry generates E, P, S.
c checks $V(E, P, S) = 1$.
generates secret a,
P on E.
generates secret b,
P on E.
computes abP.
computes abP.
and Bob convert
AES key.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature fails to formalize any of this.
Also fails to formalize analogous questions about selecting ciphers, protocols, etc.

Warmup

Extensiv
Pollard
Pohlig–H
MOV/F
SmartAS

V_1: any
public cr

re accurate
aphy.
H_V:

es E, P, S.
$V(E, P, S) = 1$.
secret a,

secret b,

abP.
abP.
convert

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature
fails to formalize any of this.
Also fails to formalize
analogous questions about
selecting ciphers, protocols, etc.

Warmup: Manipul

Extensive ECDL/E
Pollard rho breaks
Pohlig–Hellman br
MOV/FR breaks s
SmartASS breaks

V_1: any curve surv
public criteria is a

e

$= 1.$

What is V?

Which curves will public accept?

What does Jerry do?

Will he accidentally help us?

How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature fails to formalize any of this. Also fails to formalize analogous questions about selecting ciphers, protocols, etc.

Warmup: Manipulating curv

Extensive ECDL/ECDH liter

Pollard rho breaks small E,

Pohlig–Hellman breaks most

MOV/FR breaks some E,

SmartASS breaks some E, e

V_1: any curve surviving thes public criteria is acceptable.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature fails to formalize any of this. Also fails to formalize analogous questions about selecting ciphers, protocols, etc.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

What is V?
Which curves will public accept?

What does Jerry do?
Will he accidentally help us?
How robust is this protocol?

How secure is this protocol if Jerry works for us?

Traditional crypto literature
fails to formalize any of this.
Also fails to formalize
analogous questions about
selecting ciphers, protocols, etc.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these
public criteria is acceptable.

Assume that we've figured out
how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

V?

urves will public accept?

oes Jerry do?

accidentally help us?

ust is this protocol?

cure is this protocol

works for us?

nal crypto literature

formalize any of this.

s to formalize

us questions about

g ciphers, protocols, etc.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 pla

Would t

any curv

that surv

public accept?

lo?

ly help us?

protocol?

is protocol

r us?

literature

ny of this.

lize

ns about

protocols, etc.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 plausible?

Would terrorists re

any curve chosen

that survives these

cept?

?

?

ol

.

etc.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 plausible?

Would terrorists really accep
any curve chosen by Jerry
that survives these criteria?

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 plausible?

Would terrorists really accept *any* curve chosen by Jerry that survives these criteria?

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 plausible?

Would terrorists really accept *any* curve chosen by Jerry that survives these criteria?

Example showing plausibility:
French ANSSI FRP256V1 (2011)
is a random-looking curve
that survives these criteria
and has no other justification.

Warmup: Manipulating curves

Extensive ECDL/ECDH literature:
Pollard rho breaks small E,
Pohlig–Hellman breaks most E,
MOV/FR breaks some E,
SmartASS breaks some E, etc.

V_1: any curve surviving these public criteria is acceptable.

Assume that we've figured out how to break another curve E.

Jerry standardizes this curve.
Alice and Bob use it.

Is V_1 plausible?

Would terrorists really accept *any* curve chosen by Jerry that survives these criteria?

Example showing plausibility:
French ANSSI FRP256V1 (2011) is a random-looking curve that survives these criteria and has no other justification.

Earlier example:
Chinese OSCCA SM2 (2010).

o: Manipulating curves

e ECDL/ECDH literature:
rho breaks small E,
Hellman breaks most E,
R breaks some E,
SS breaks some E, etc.

curve surviving these
riteria is acceptable.

that we've figured out
break another curve E.

andardizes this curve.
d Bob use it.

Is V_1 plausible?

Would terrorists really accept
any curve chosen by Jerry
that survives these criteria?

Example showing plausibility:
French ANSSI FRP256V1 (2011)
is a random-looking curve
that survives these criteria
and has no other justification.

Earlier example:
Chinese OSCCA SM2 (2010).

Manipul

V_2: curv
criteria,
a "seed"

Example
"selectin
verifiably
SEC 2 1
random
some ad
features'
be prede
186-2 (2
Certicom

lating curves

ECDH literature:
small E,
reaks most E,
some E,
some E, etc.

viving these
cceptable.

e figured out
her curve E.

this curve.
it.

Is V_1 plausible?

Would terrorists really accept
any curve chosen by Jerry
that survives these criteria?

Example showing plausibility:
French ANSSI FRP256V1 (2011)
is a random-looking curve
that survives these criteria
and has no other justification.

Earlier example:
Chinese OSCCA SM2 (2010).

Manipulating seed

V_2: curve must sa
criteria, *and* Jerry
a "seed" s such th

Examples: ANSI X
"selecting an ellipt
verifiably at rando
SEC 2 1.0 (2000)
random parameter
some additional co
features"—"param
be predetermined"
186-2 (2000); ANS
Certicom SEC 2 2

ves

rature:

t E,

etc.

e

out

E.

e.

Is V_1 plausible?

Would terrorists really accept
any curve chosen by Jerry
that survives these criteria?

Example showing plausibility:
French ANSSI FRP256V1 (2011)
is a random-looking curve
that survives these criteria
and has no other justification.

Earlier example:
Chinese OSCCA SM2 (2010).

Manipulating seeds

V_2: curve must satisfy the p
criteria, *and* Jerry must prov
a "seed" s such that $E = H$

Examples: ANSI X9.62 (199
"selecting an elliptic curve
verifiably at random"; Certic
SEC 2 1.0 (2000) "verifiably
random parameters offer
some additional conservative
features"—"parameters cann
be predetermined"; NIST FI
186-2 (2000); ANSI X9.63 (
Certicom SEC 2 2.0 (2010).

Is V_1 plausible?

Would terrorists really accept *any* curve chosen by Jerry that survives these criteria?

Example showing plausibility: French ANSSI FRP256V1 (2011) is a random-looking curve that survives these criteria and has no other justification.

Earlier example: Chinese OSCCA SM2 (2010).

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

usible?

errorists really accept
ve chosen by Jerry
vives these criteria?

e showing plausibility:
ANSSI FRP256V1 (2011)
dom-looking curve
vives these criteria
no other justification.

xample:
OSCCA SM2 (2010).

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

What ex

NIST de
$y^2 = x^3$
$b^2c = -$
hash is S

eally accept
by Jerry
e criteria?

plausibility:
P256V1 (2011)
ng curve
e criteria
justification.

M2 (2010).

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

What exactly is H

NIST defines curv
$y^2 = x^3 - 3x + b$
$b^2c = -27$; c is a
hash is SHA-1 co

pt

y:
2011)

n.

).

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

What exactly is H?

NIST defines curve E as $y^2 = x^3 - 3x + b$ where $b^2c = -27$; c is a hash of s hash is SHA-1 concatenation

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

What exactly is H?

NIST defines curve E as $y^2 = x^3 - 3x + b$ where $b^2c = -27$; c is a hash of s; hash is SHA-1 concatenation.

Manipulating seeds

V_2: curve must satisfy the public criteria, *and* Jerry must provide a "seed" s such that $E = H(s)$.

Examples: ANSI X9.62 (1999) "selecting an elliptic curve verifiably at random"; Certicom SEC 2 1.0 (2000) "verifiably random parameters offer some additional conservative features"—"parameters cannot be predetermined"; NIST FIPS 186-2 (2000); ANSI X9.63 (2001); Certicom SEC 2 2.0 (2010).

What exactly is H?

NIST defines curve E as $y^2 = x^3 - 3x + b$ where $b^2c = -27$; c is a hash of s; hash is SHA-1 concatenation.

But clearly public will accept other choices of H.

Examples: Brainpool (2005) uses $c = g^3/h^2$ where g and h are separate hashes. NIST FIPS 186-4 (2013) requires an "approved hash function, as specified in FIPS 180"; no longer allows SHA-1!

ating seeds

e must satisfy the public
and Jerry must provide
s such that $E = H(s)$.

es: ANSI X9.62 (1999)
ng an elliptic curve
y at random"; Certicom
.0 (2000) "verifiably
parameters offer
ditional conservative
"—"parameters cannot
etermined"; NIST FIPS
2000); ANSI X9.63 (2001);
n SEC 2 2.0 (2010).

What exactly is H?

NIST defines curve E as
$y^2 = x^3 - 3x + b$ where
$b^2c = -27$; c is a hash of s;
hash is SHA-1 concatenation.

But clearly public will accept
other choices of H.

Examples: Brainpool (2005)
uses $c = g^3/h^2$ where
g and h are separate hashes.
NIST FIPS 186-4 (2013) requires
an "approved hash function, as
specified in FIPS 180";
no longer allows SHA-1!

1999 Sc
possibilit
of all cu
structure
but we c
generate
until the
one of 't
get us to

s

tisfy the public
must provide
hat $E = H(s)$.

K9.62 (1999)
tic curve
m"; Certicom
"verifiably
rs offer
onservative
neters cannot
; NIST FIPS
SI X9.63 (2001);
.0 (2010).

What exactly is H?

NIST defines curve E as
$y^2 = x^3 - 3x + b$ where
$b^2c = -27$; c is a hash of s;
hash is SHA-1 concatenation.

But clearly public will accept
other choices of H.

Examples: Brainpool (2005)
uses $c = g^3/h^2$ where
g and h are separate hashes.
NIST FIPS 186-4 (2013) requires
an "approved hash function, as
specified in FIPS 180";
no longer allows SHA-1!

1999 Scott: "Cons
possibility that on
of all curves have
structure that 'the
but we don't. The
generate a million
until they find one
one of 'their' curv
get us to use them

ublic
vide
(s).
99)
com
y
e
not
PS
2001);

What exactly is H?

NIST defines curve E as
$y^2 = x^3 - 3x + b$ where
$b^2c = -27$; c is a hash of s;
hash is SHA-1 concatenation.

But clearly public will accept
other choices of H.

Examples: Brainpool (2005)
uses $c = g^3/h^2$ where
g and h are separate hashes.
NIST FIPS 186-4 (2013) requires
an "approved hash function, as
specified in FIPS 180";
no longer allows SHA-1!

1999 Scott: "Consider now
possibility that one in a mill
of all curves have an exploit
structure that 'they' know a
but we don't. Then 'they' s
generate a million random s
until they find one that gene
one of 'their' curves. Then t
get us to use them."

What exactly is H?

NIST defines curve E as
$y^2 = x^3 - 3x + b$ where
$b^2c = -27$; c is a hash of s;
hash is SHA-1 concatenation.

But clearly public will accept
other choices of H.

Examples: Brainpool (2005)
uses $c = g^3/h^2$ where
g and h are separate hashes.
NIST FIPS 186-4 (2013) requires
an "approved hash function, as
specified in FIPS 180";
no longer allows SHA-1!

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

What exactly is H?

NIST defines curve E as
$y^2 = x^3 - 3x + b$ where
$b^2c = -27$; c is a hash of s;
hash is SHA-1 concatenation.

But clearly public will accept
other choices of H.

Examples: Brainpool (2005)
uses $c = g^3/h^2$ where
g and h are separate hashes.
NIST FIPS 186-4 (2013) requires
an "approved hash function, as
specified in FIPS 180";
no longer allows SHA-1!

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

New: Optimized this computation using Keccak on cluster of 41 GTX780 GPUs. In 7 hours found "secure+twist-secure" $b =$ `0x`
`BADA55EC`D8BBEAD3ADD6C534F92197DE
B47FCEB9BE7E0E702A8D1DD56B5D0B0C.

xactly is H?

efines curve E as
$-3x+b$ where
-27; c is a hash of s;
SHA-1 concatenation.

rly public will accept
oices of H.

es: Brainpool (2005)
$= g^3/h^2$ where
are separate hashes.
PS 186-4 (2013) requires
roved hash function, as
l in FIPS 180";
er allows SHA-1!

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

New: Optimized this computation using Keccak on cluster of 41 GTX780 GPUs. In 7 hours found "secure+twist-secure" $b =$ 0x
BADA55ECD8BBEAD3ADD6C534F92197DE
B47FCEB9BE7E0E702A8D1DD56B5D0B0C.

Manipul

Brainpo
"The ch
from wh
paramet
not mot
part of t
open. . .

Verifiab

The [Bra
generate
manner
generate
compreh

?

e E as
where
hash of s;
ncatenation.

will accept
l.

ool (2005)
here
ate hashes.
(2013) requires
n function, as
180";
HA-1!

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

New: Optimized this computation using Keccak on cluster of 41 GTX780 GPUs. In 7 hours found "secure+twist-secure" $b =$ 0x
BADA55ECD8BBEAD3ADD6C534F92197DE
B47FCEB9BE7E0E702A8D1DD56B5D0B0C.

Manipulating NUM

Brainpool standard
"The choice of the
from which the [N
parameters have b
not motivated leav
part of the security
open. . . .

Verifiably pseudo

The [Brainpool] cu
generated in a pse
manner using seed
generated in a syst
comprehensive way

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

New: Optimized this computation using Keccak on cluster of 41 GTX780 GPUs. In 7 hours found "secure+twist-secure" $b =$ 0x BADA55ECD8BBEAD3ADD6C534F92197DE B47FCEB9BE7E0E702A8D1DD56B5D0B0C.

Manipulating NUMS numbe

Brainpool standard:
"The choice of the seeds from which the [NIST] curve parameters have been derive not motivated leaving an ess part of the security analysis open. . . .

Verifiably pseudo-random

The [Brainpool] curves shall generated in a pseudo-rando manner using seeds that are generated in a systematic an comprehensive way."

1999 Scott: "Consider now the possibility that one in a million of all curves have an exploitable structure that 'they' know about, but we don't. Then 'they' simply generate a million random seeds until they find one that generates one of 'their' curves. Then they get us to use them."

New: Optimized this computation using Keccak on cluster of 41 GTX780 GPUs. In 7 hours found "secure+twist-secure" $b =$ 0x BADA55ECD8BBEAD3ADD6C534F92197DE B47FCEB9BE7E0E702A8D1DD56B5D0B0C.

Manipulating NUMS numbers

Brainpool standard:
"The choice of the seeds from which the [NIST] curve parameters have been derived is not motivated leaving an essential part of the security analysis open. . . .
Verifiably pseudo-random.
The [Brainpool] curves shall be generated in a pseudo-random manner using seeds that are generated in a systematic and comprehensive way."

ott: “Consider now the
ty that one in a million
rves have an exploitable
e that ‘they’ know about,
don’t. Then ‘they’ simply
e a million random seeds
ey find one that generates
their’ curves. Then they
o use them.”

ptimized this computation
eccak on cluster of 41
GPUs. In 7 hours found
+twist-secure” $b =$ 0x
CD8BBEAD3ADD6C534F92197DE
9BE7E0E702A8D1DD56B5D0B0C.

Manipulating NUMS numbers

Brainpool standard:
“The choice of the seeds
from which the [NIST] curve
parameters have been derived is
not motivated leaving an essential
part of the security analysis
open. . . .

Verifiably pseudo-random.
The [Brainpool] curves shall be
generated in a pseudo-random
manner using seeds that are
generated in a systematic and
comprehensive way.”

Wikiped
nothing
are any
construc
of hidde

Microsof
“generat
from the

Albertin
Mendel–
hashing”
in hash
expected
nothing-

sider now the
e in a million
an exploitable
ey' know about,
en 'they' simply
random seeds
e that generates
es. Then they
n."

his computation
luster of 41
n 7 hours found
ure" $b =$ 0x
ADD6C534F92197DE
2A8D1DD56B5D0B0C.

Manipulating NUMS numbers

Brainpool standard:
"The choice of the seeds
from which the [NIST] curve
parameters have been derived is
not motivated leaving an essential
part of the security analysis
open. . . .
Verifiably pseudo-random.
The [Brainpool] curves shall be
generated in a pseudo-random
manner using seeds that are
generated in a systematic and
comprehensive way."

Wikipedia: "In cry
nothing up my sl
are any numbers w
construction, are a
of hidden properti

Microsoft "NUMS
"generated determ
from the security

Albertini–Aumasso
Mendel–Schläffer
hashing" (2014):
in hash functions a
expected to be ide
nothing-up-your-sl

the
ion
able
bout,
imply
eeds
erates
they

tation
41
found
0x
92197DE
B5D0B0C.

Manipulating NUMS numbers

Brainpool standard:
"The choice of the seeds
from which the [NIST] curve
parameters have been derived is
not motivated leaving an essential
part of the security analysis
open. . . .
Verifiably pseudo-random.
The [Brainpool] curves shall be
generated in a pseudo-random
manner using seeds that are
generated in a systematic and
comprehensive way."

Wikipedia: "In cryptography
nothing up my sleeve num
are any numbers which, by t
construction, are above susp
of hidden properties."

Microsoft "NUMS" curves (
"generated deterministically
from the security level".

Albertini–Aumasson–Eichlse
Mendel–Schläffer "Malicious
hashing" (2014): "constants
in hash functions are normal
expected to be identifiable a
nothing-up-your-sleeve numb

Manipulating NUMS numbers

Brainpool standard:
"The choice of the seeds
from which the [NIST] curve
parameters have been derived is
not motivated leaving an essential
part of the security analysis
open. . . .
Verifiably pseudo-random.
The [Brainpool] curves shall be
generated in a pseudo-random
manner using seeds that are
generated in a systematic and
comprehensive way."

Wikipedia: "In cryptography,
nothing up my sleeve numbers
are any numbers which, by their
construction, are above suspicion
of hidden properties."

Microsoft "NUMS" curves (2014):
"generated deterministically
from the security level".

Albertini–Aumasson–Eichlseder–
Mendel–Schläffer "Malicious
hashing" (2014): "constants
in hash functions are normally
expected to be identifiable as
nothing-up-your-sleeve numbers".

ating NUMS numbers

ol standard:
oice of the seeds
ich the [NIST] curve
ers have been derived is
ivated leaving an essential
he security analysis
.
ly pseudo-random.
ainpool] curves shall be
ed in a pseudo-random
using seeds that are
ed in a systematic and
ensive way."

Wikipedia: "In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties."

Microsoft "NUMS" curves (2014): "generated deterministically from the security level".

Albertini–Aumasson–Eichlseder–Mendel–Schläffer "Malicious hashing" (2014): "constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers".

New: W
curve "E
with a B

MS numbers

d:
e seeds
IST] curve
been derived is
ving an essential
y analysis

o-random.
urves shall be
eudo-random
ls that are
tematic and
y."

Wikipedia: "In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties."

Microsoft "NUMS" curves (2014): "generated deterministically from the security level".

Albertini–Aumasson–Eichlseder–Mendel–Schläffer "Malicious hashing" (2014): "constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers".

New: We generate
curve "BADA55-V
with a Brainpool-I

ers

e
ed is
sential

be
om

nd

Wikipedia: "In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties."

Microsoft "NUMS" curves (2014): "generated deterministically from the security level".

Albertini–Aumasson–Eichlseder–Mendel–Schläffer "Malicious hashing" (2014): "constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers".

New: We generated a BADA
curve "BADA55-VPR-224"
with a Brainpool-like explan

Wikipedia: “In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties.”

Microsoft “NUMS” curves (2014): “generated deterministically from the security level”.

Albertini–Aumasson–Eichlseder–Mendel–Schläffer “Malicious hashing” (2014): “constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers”.

New: We generated a BADA55 curve “BADA55-VPR-224” with a Brainpool-like explanation.

Wikipedia: "In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties."

Microsoft "NUMS" curves (2014): "generated deterministically from the security level".

Albertini–Aumasson–Eichlseder–Mendel–Schläffer "Malicious hashing" (2014): "constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers".

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Wikipedia: "In cryptography, **nothing up my sleeve numbers** are any numbers which, by their construction, are above suspicion of hidden properties."

Microsoft "NUMS" curves (2014): "generated deterministically from the security level".

Albertini–Aumasson–Eichlseder–Mendel–Schläffer "Malicious hashing" (2014): "constants in hash functions are normally expected to be identifiable as nothing-up-your-sleeve numbers".

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Example of underlying flexibility: Brainpool generates seeds from exp(1) and primes from arctan(1); MD5 generates constants from sin(1); BADA55-VPR-224 generated a seed from cos(1).

ia: "In cryptography,
up my sleeve numbers
numbers which, by their
ction, are above suspicion
n properties."

ft "NUMS" curves (2014):
ted deterministically
e security level".

i–Aumasson–Eichlseder–
-Schläffer "Malicious
" (2014): "constants
functions are normally
d to be identifiable as
up-your-sleeve numbers".

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Example of underlying flexibility: Brainpool generates seeds from exp(1) and primes from arctan(1); MD5 generates constants from sin(1); BADA55-VPR-224 generated a seed from cos(1).

Most ma
was draw

How to
manipul
a white

Daniel J
Tung Ch
Chitchan
Andreas
Tanja La
Ruben N
Christine

safecur
/bada55

yptography,
eeve numbers
which, by their
above suspicion
es."

" curves (2014):
ninistically
evel".

on–Eichlseder–
"Malicious
"constants
are normally
entifiable as
eeve numbers".

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Example of underlying flexibility: Brainpool generates seeds from exp(1) and primes from arctan(1); MD5 generates constants from sin(1); BADA55-VPR-224 generated a seed from cos(1).

Most material in t
was drawn from th

How to
manipulate curve s
a white paper for

Daniel J. Bernstei
Tung Chou
Chitchanok Chuen
Andreas Hülsing
Tanja Lange
Ruben Niederhage
Christine van Vred

safecurves.cr.y
/bada55.html

y,
nbers
their
picion

2014):

der–
s
s
lly
ns
bers".

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Example of underlying flexibility: Brainpool generates seeds from exp(1) and primes from arctan(1); MD5 generates constants from sin(1); BADA55-VPR-224 generated a seed from cos(1).

Most material in this talk was drawn from this paper:

How to manipulate curve standards: a white paper for the black

Daniel J. Bernstein
Tung Chou
Chitchanok Chuengsatiansup
Andreas Hülsing
Tanja Lange
Ruben Niederhagen
Christine van Vredendaal

safecurves.cr.yp.to/bada55.html

New: We generated a BADA55 curve "BADA55-VPR-224" with a Brainpool-like explanation.

We actually generated >1000000 curves, each having a Brainpool-like explanation.

Example of underlying flexibility: Brainpool generates seeds from exp(1) and primes from arctan(1); MD5 generates constants from sin(1); BADA55-VPR-224 generated a seed from cos(1).

Most material in this talk was drawn from this paper:

How to manipulate curve standards: a white paper for the black hat

Daniel J. Bernstein
Tung Chou
Chitchanok Chuengsatiansup
Andreas Hülsing
Tanja Lange
Ruben Niederhagen
Christine van Vredendaal

safecurves.cr.yp.to/bada55.html