Understanding brute force

D. J. Bernstein

Thanks to:
University of Illinois at Chicago
NSF CCR–9983950
Alfred P. Sloan Foundation

Cryptanalyst wants to find secret 128-bit AES key k, given $\text{AES}_k(0)$.

He builds an attack machine.

Machine 1: His desktop PC, searching through n possibilities for k.

Machine costs $\approx 2^9$ dollars; takes $\approx n/2^{22}$ seconds; succeeds with chance $n/2^{128}$.

te force

is at Chicago
0
oundation

Cryptanalyst wants to find secret 128-bit AES key k, given $\text{AES}_k(0)$.

He builds an attack machine.

Machine 1: His desktop PC, searching through n possibilities for k.

Machine costs $\approx 2^9$ dollars; takes $\approx n/2^{22}$ seconds; succeeds with chance $n/2^{128}$.

This is a silly attac
The cryptanalyst h

Machine 2: p desk
each searching thr
n possibilities for

Machine costs $\approx$ 2
takes $\approx n/2^{22}$ sec
succeeds with char

Same keys/dollar-s
Same chance/doll
But larger chance!

Cryptanalyst wants to find secret 128-bit AES key k, given $\text{AES}_k(0)$.

He builds an attack machine.

Machine 1: His desktop PC, searching through n possibilities for k.

Machine costs $\approx 2^9$ dollars; takes $\approx n/2^{22}$ seconds; succeeds with chance $n/2^{128}$.

This is a silly attack machine. The cryptanalyst has more money.

Machine 2: p desktop PCs, each searching through n possibilities for k.

Machine costs $\approx 2^9 p$ dollars; takes $\approx n/2^{22}$ seconds; succeeds with chance $pn/2^{128}$.

Same keys/dollar-second: 2^{13}.
Same chance/dollar-second: 2^{-115}.
But larger chance!

s to find
S key k,

ck machine.

esktop PC,

k.

2^9 dollars;
onds;
nce $n/2^{128}$.

This is a silly attack machine.
The cryptanalyst has more money.

Machine 2: p desktop PCs,
each searching through
n possibilities for k.

Machine costs $\approx 2^9 p$ dollars;
takes $\approx n/2^{22}$ seconds;
succeeds with chance $pn/2^{128}$.

Same keys/dollar-second: 2^{13}.
Same chance/dollar-second: 2^{-115}.
But larger chance!

This is a silly attac
Only a tiny part of
is doing anything u

Machine 3: p tiny
each searching thr
n possibilities for

AES circuit, in bul
is orders of magnit
less expensive thar
allowing much larg
Cost ratio grows w

Recall DES Cracke
2^{19} keys/dollar-sec

This is a silly attack machine.
The cryptanalyst has more money.

Machine 2: p desktop PCs, each searching through n possibilities for k.

Machine costs $\approx 2^9 p$ dollars; takes $\approx n/2^{22}$ seconds; succeeds with chance $pn/2^{128}$.

Same keys/dollar-second: 2^{13}.
Same chance/dollar-second: 2^{-115}.
But larger chance!

This is a silly attack machine.
Only a tiny part of the PC is doing anything useful.

Machine 3: p tiny AES circuits, each searching through n possibilities for k.

AES circuit, in bulk, is orders of magnitude less expensive than PC, allowing much larger p.
Cost ratio grows with PC size!

Recall DES Cracker: in 1997, 2^{19} keys/dollar-second.

ck machine.
has more money.

ktop PCs,
ough
k.

$2^9 p$ dollars;
onds;
nce $pn/2^{128}$.

second: 2^{13}.
ar-second: 2^{-115}.

This is a silly attack machine.
Only a tiny part of the PC
is doing anything useful.

Machine 3: p tiny AES circuits,
each searching through
n possibilities for k.

AES circuit, in bulk,
is orders of magnitude
less expensive than PC,
allowing much larger p.
Cost ratio grows with PC size!

Recall DES Cracker: in 1997,
2^{19} keys/dollar-second.

This is still silly *if*
cryptanalyst is act
many keys $k_1, k_2,$

Complicated but s
brute-force key-sea
handles $\approx \sqrt{p}$ key
using rainbow tabl
$\approx p$ using distingu

Similar time, price
Conjecturally $\approx pn$
of success for ever
distinguished poin

This is a silly attack machine. Only a tiny part of the PC is doing anything useful.

Machine 3: p tiny AES circuits, each searching through n possibilities for k.

AES circuit, in bulk, is orders of magnitude less expensive than PC, allowing much larger p. Cost ratio grows with PC size!

Recall DES Cracker: in 1997, 2^{19} keys/dollar-second.

This is still silly *if* cryptanalyst is actually attacking many keys $k_1, k_2, k_3, \ldots$.

Complicated but standard parallel brute-force key-search machine handles $\approx \sqrt{p}$ keys at once using rainbow tables, or $\approx p$ using distinguished points.

Similar time, price to one key. Conjecturally $\approx pn/2^{128}$ chance of success for every key; distinguished points, slightly lower.

ck machine.
f the PC
useful.

AES circuits,
ough
k.

lk,
tude
n PC,
ger p.
vith PC size!

er: in 1997,
cond.

This is still silly *if*
cryptanalyst is actually attacking
many keys $k_1, k_2, k_3, \ldots$.

Complicated but standard parallel
brute-force key-search machine
handles $\approx \sqrt{p}$ keys at once
using rainbow tables, or
$\approx p$ using distinguished points.

Similar time, price to one key.
Conjecturally $\approx pn/2^{128}$ chance
of success for every key;
distinguished points, slightly lower.

Is this acceptable s
If not, what do we

Option 1: Input-sp
to stop many-keys
"Use a large rando
Heavy costs (usua
limited benefits.

Option 2: Use 32-
"Randomness in k
Smaller costs; larg

See paper for furth
`http://cr.yp.to`
`/papers.html#br`

This is still silly *if* cryptanalyst is actually attacking many keys $k_1, k_2, k_3, \ldots$.

Complicated but standard parallel brute-force key-search machine handles $\approx \sqrt{p}$ keys at once using rainbow tables, or $\approx p$ using distinguished points.

Similar time, price to one key. Conjecturally $\approx pn/2^{128}$ chance of success for every key; distinguished points, slightly lower.

Is this acceptable security? If not, what do we do?

Option 1: Input-space separation, to stop many-keys attacks. "Use a large random nonce." Heavy costs (usually understated); limited benefits.

Option 2: Use 32-byte keys. "Randomness in key, not nonce." Smaller costs; larger benefits.

See paper for further analysis: `http://cr.yp.to/papers.html#bruteforce`

ually attacking
$k_3, \ldots$.

tandard parallel
arch machine
s at once
es, or
ished points.

to one key.
$n/2^{128}$ chance
y key;
ts, slightly lower.

Is this acceptable security?
If not, what do we do?

Option 1: Input-space separation,
to stop many-keys attacks.
"Use a large random nonce."
Heavy costs (usually understated);
limited benefits.

Option 2: Use 32-byte keys.
"Randomness in key, not nonce."
Smaller costs; larger benefits.

See paper for further analysis:
`http://cr.yp.to`
`/papers.html#bruteforce`

<u>Basic cryptanalytic</u>

A new attack is po
it takes *less* time
than standard brut
at the *same* price
with the *same* suc

Most papers get th
Example: The atta
9 rounds of 256-bi
had larger price an
complete brute-for
through all 2^{256} ke

Is this acceptable security?
If not, what do we do?

Option 1: Input-space separation, to stop many-keys attacks. "Use a large random nonce." Heavy costs (usually understated); limited benefits.

Option 2: Use 32-byte keys. "Randomness in key, not nonce." Smaller costs; larger benefits.

See paper for further analysis: `http://cr.yp.to/papers.html#bruteforce`

Basic cryptanalytic economics

A new attack is pointless unless it takes *less* time than standard brute-force machine at the *same* price with the *same* success chance.

Most papers get this wrong. Example: The attack "breaking" 9 rounds of 256-bit Serpent had larger price and time than a complete brute-force search through all 2^{256} keys.