

# Experiences with Model Inference Assisted Fuzzing

# Experiences with Model Inference Assisted Fuzzing

or

# Experiences with Model Inference Assisted Fuzzing

or

How Blue Sky Model Inference Research  
Turned Into  
Yet Another Fuzzing Experiment

# Experiences with Model Inference Assisted Fuzzing

or

How Blue Sky Model Inference Research  
Turned Into

**THE BEST** Fuzzing Experiment

# Experiences with Model Inference Assisted Fuzzing

or

How Blue Sky Model Inference Research  
Turned Into

**THE BEST** Fuzzing Experiment

**LIKE, EVER!**

# 1996

## OUSPG

Oulu University Secure Programming Group  
(University of Finland Secure Programming Group)  
(Oslo University Security Programming Group)

# 1999

---

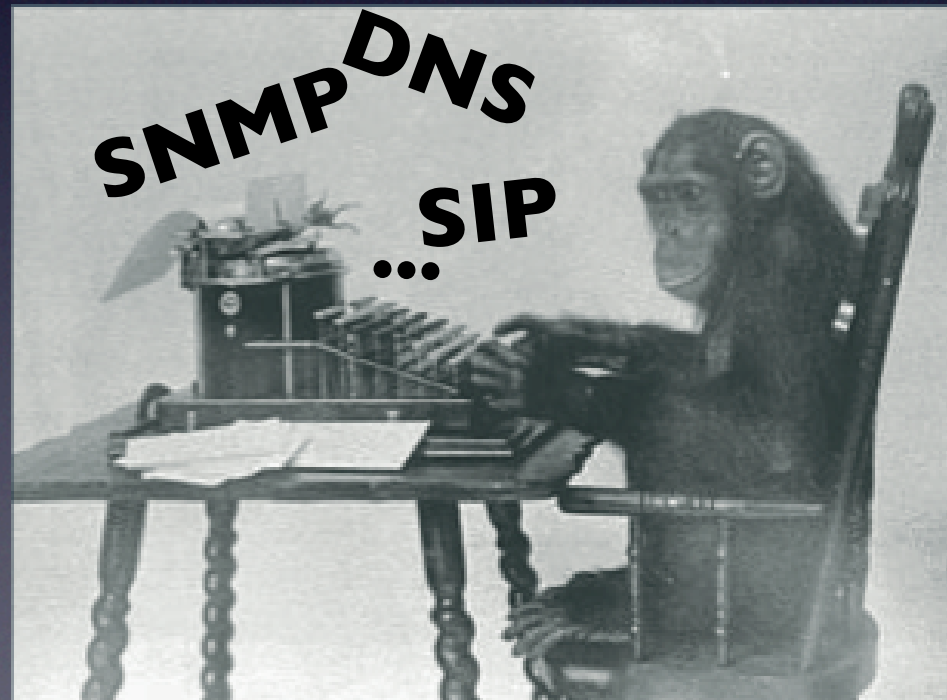
## PROTOS



# 1999

---

## PROTOS





“Gee, many protocols have same kinds of fields.”

“Those fields are kind of building blocks of protocols.”

**WARNING: A FLAWED ANALOGY FOLLOWS.**

“Kind of like genes.”

# 2004

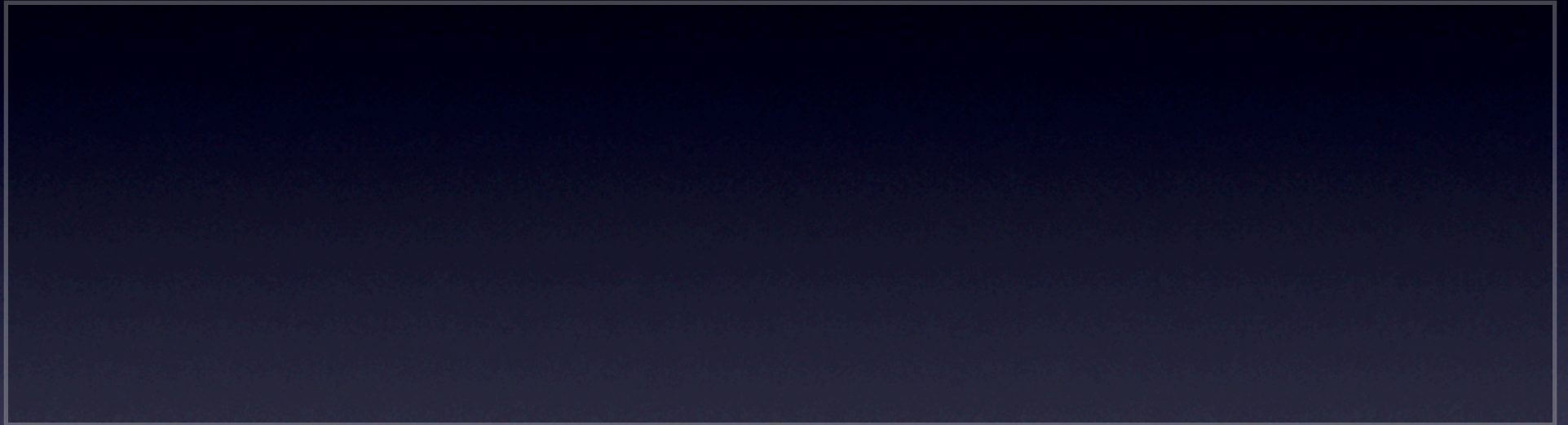
## PROTOS Genome

Harvest “protocol genes”.

Automate by stealing algorithms from bioinformatics and applying them to raw data samples.

Use the genes for nefarious purposes.

# Testing the waters: REGEXPERT



# Testing the waters: REGEXPERT

Sequence alignment algorithm

# Testing the waters: REGEXPERT

Sequence alignment algorithm  
+ GIF files



# Testing the waters: REGEXPERT

Sequence alignment algorithm

+

GIF files

+

Fairy dust

# Testing the waters: REGEXPERT

Sequence alignment algorithm

+ GIF files

+ Fairy dust

---

= GIF87a.{1}\^C.{1}\^B.{1}\000\000\000\000...

# Testing the waters: REGEXPERT

Sequence alignment algorithm  
+ GIF files  
+ Fairy dust  

---

= `GIF87a.{1}\^C.{1}\^B.{1}\000\000\000\000...`


For more dramatic effect, see

**Marshall Beddoe &  
Protocol Informatics**

“We need more POWER! Kreegah! Tarzan bundolo!”

# By the power of Turing: Functional genes

```
(let-structure
  ((ip-version (integer 4))
   (header-length (integer 4))
   (service-type byte)
   (total-length (integer 16))
   (identification (integer 16))
   (skip zero-bit)
   (DF-bit bit)
   (MF-bit bit)
   (fragment-offset (integer 13))
   (time-to-live byte)
   (protocol byte)
   (header-checksum (integer 16))
   (source-address word)
   (dest-address word)
   (options
    (repeat (- header-length 5) word))
   (payload
    (repeat
```



IPv4 header  
“genome”

# Experts-Controller

## Controller

Raw data

mckskndnjABBAsm

fdsmABBAaa

on (integer 4))  
ggggmf-length (integer 4  
8888 ice-type byte)  
total-length (integer 16  
(identification (integer  
(skip zero-bit)  
(DF-bit bit)  
(MF-bit bit)

Big picture

# Experts-Controller

Shared substring expert

mckskndnj**ABBA**smdks

fdsm**ABBA**aazdfsdxxxx

ggggmfk12345**ABBA**..!fd

“Whatever” expert

Controller

Raw data

~~mckskndnjABBAspm~~

~~fdsmABBAaaz~~

~~on (integer 4))~~

~~ggggmf/length (integer 4~~

~~8888/ice-type byte)~~

~~total-length (integer 16~~

~~(identification (integer~~

~~(skip zero-bit)~~

~~(DF-bit bit)~~

~~(MF-bit bit)~~

Big picture

Gene pool expert

length-  
prefixed  
string

MD5

SHA1

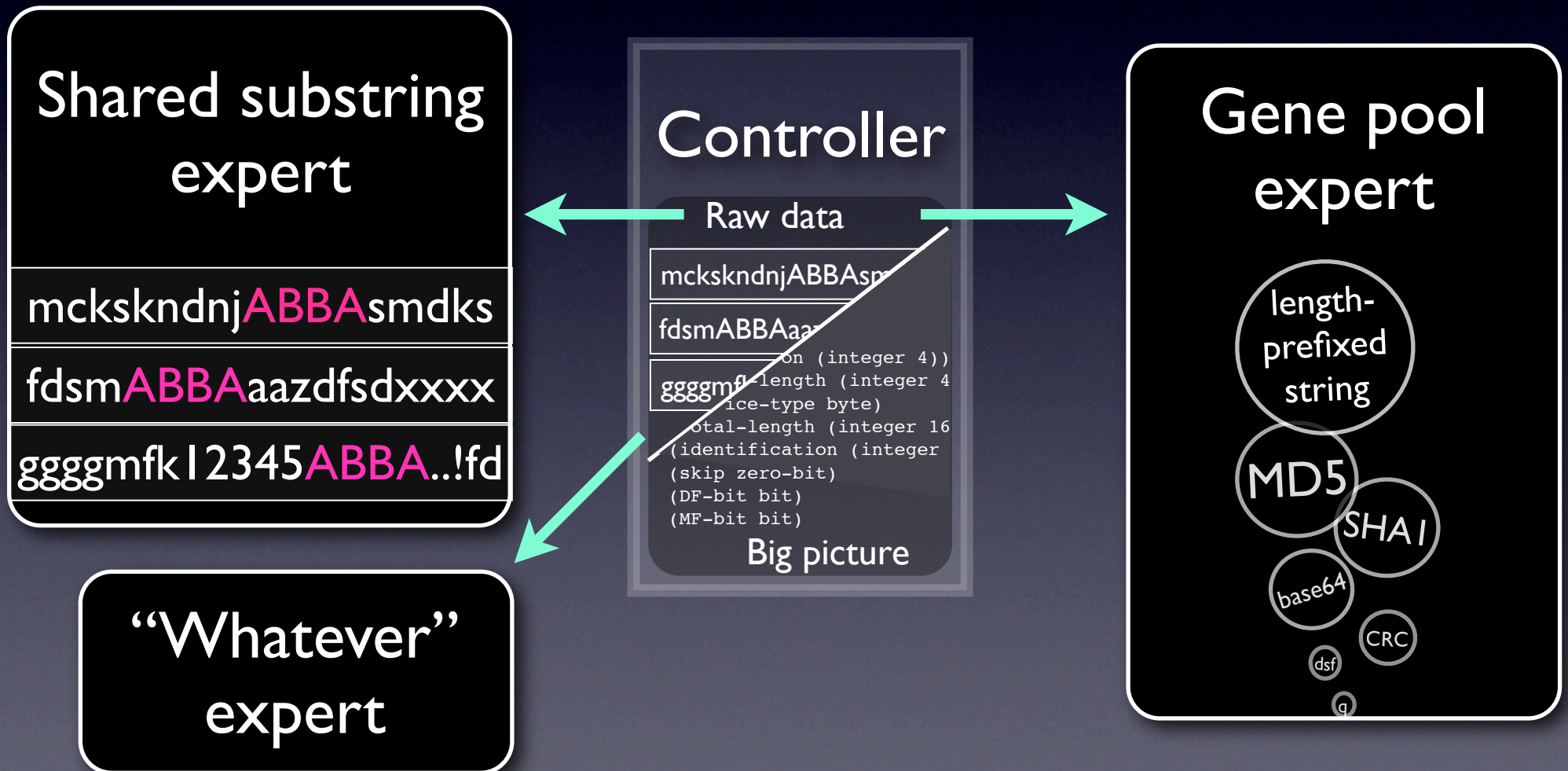
base64

CRC

dsf

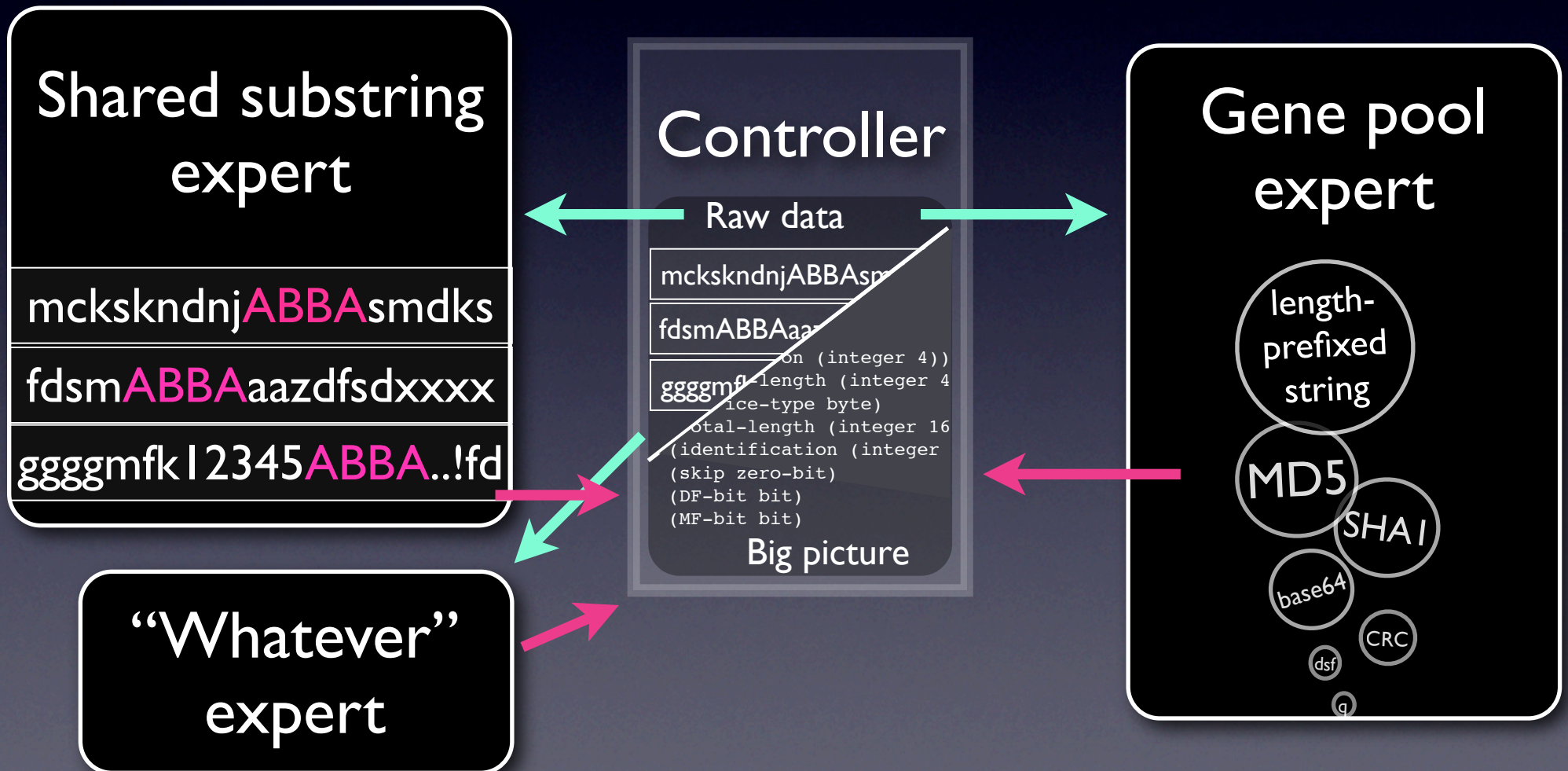
q

# Experts-Controller



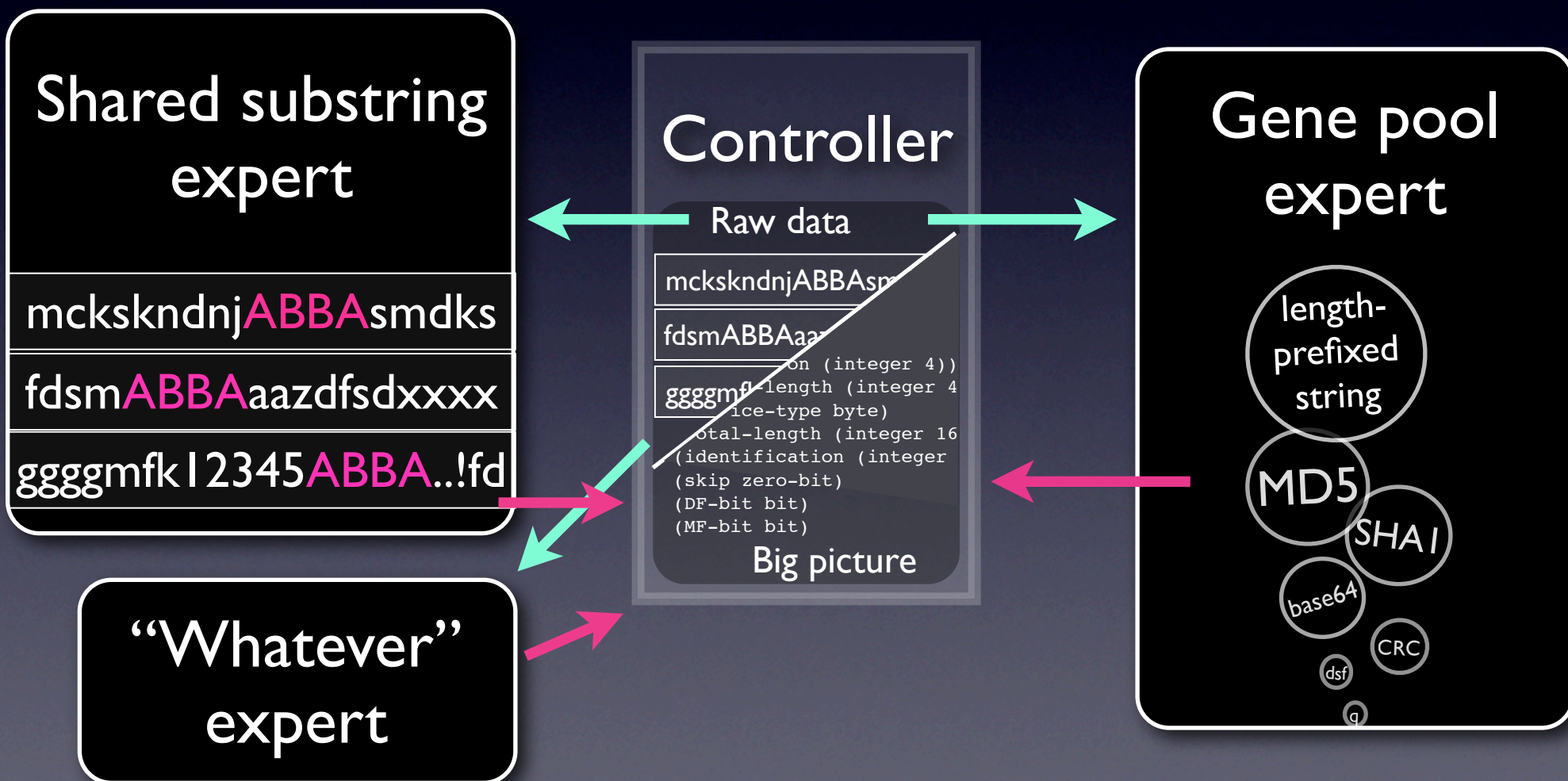


# Experts-Controller

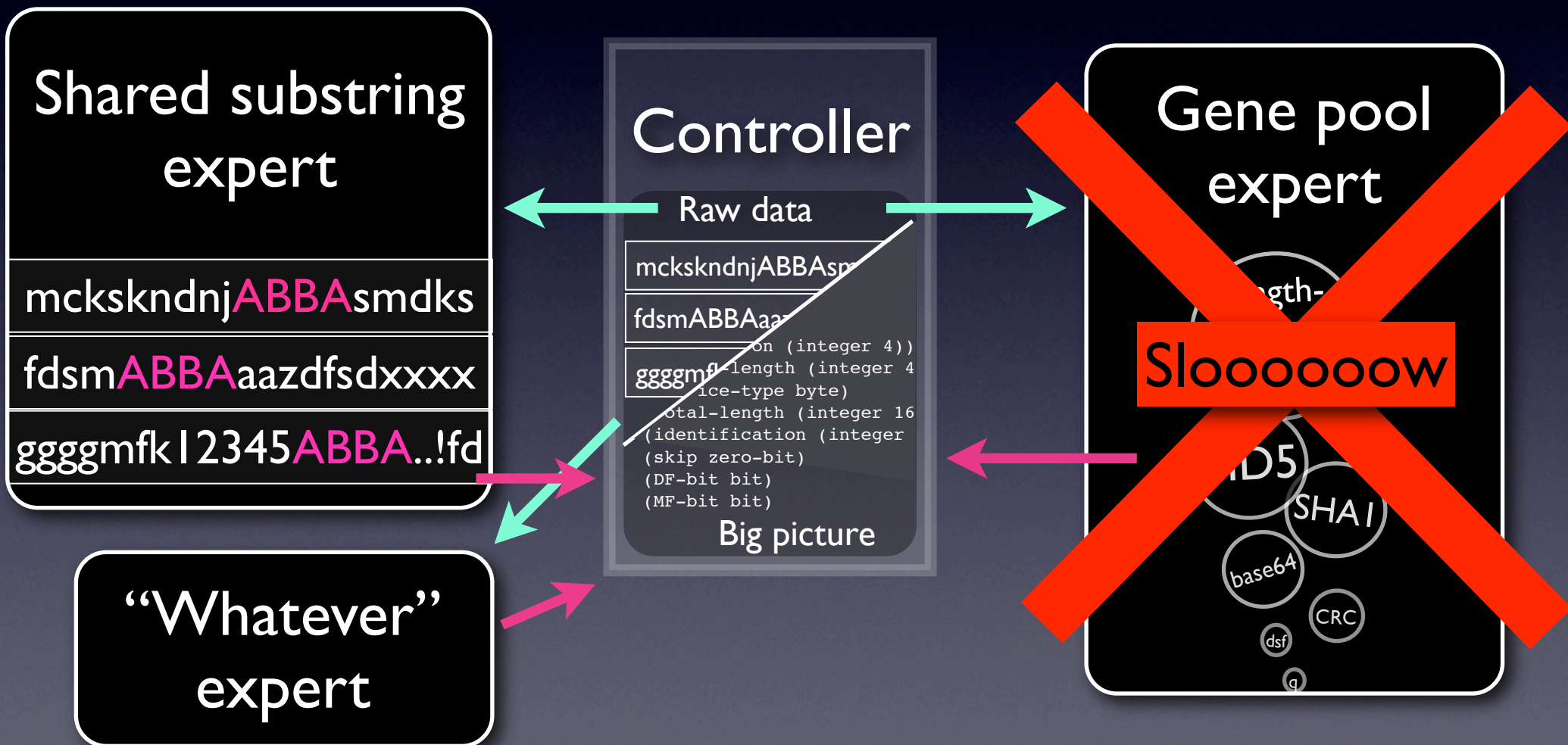


“So...You have a machine generated spec.”

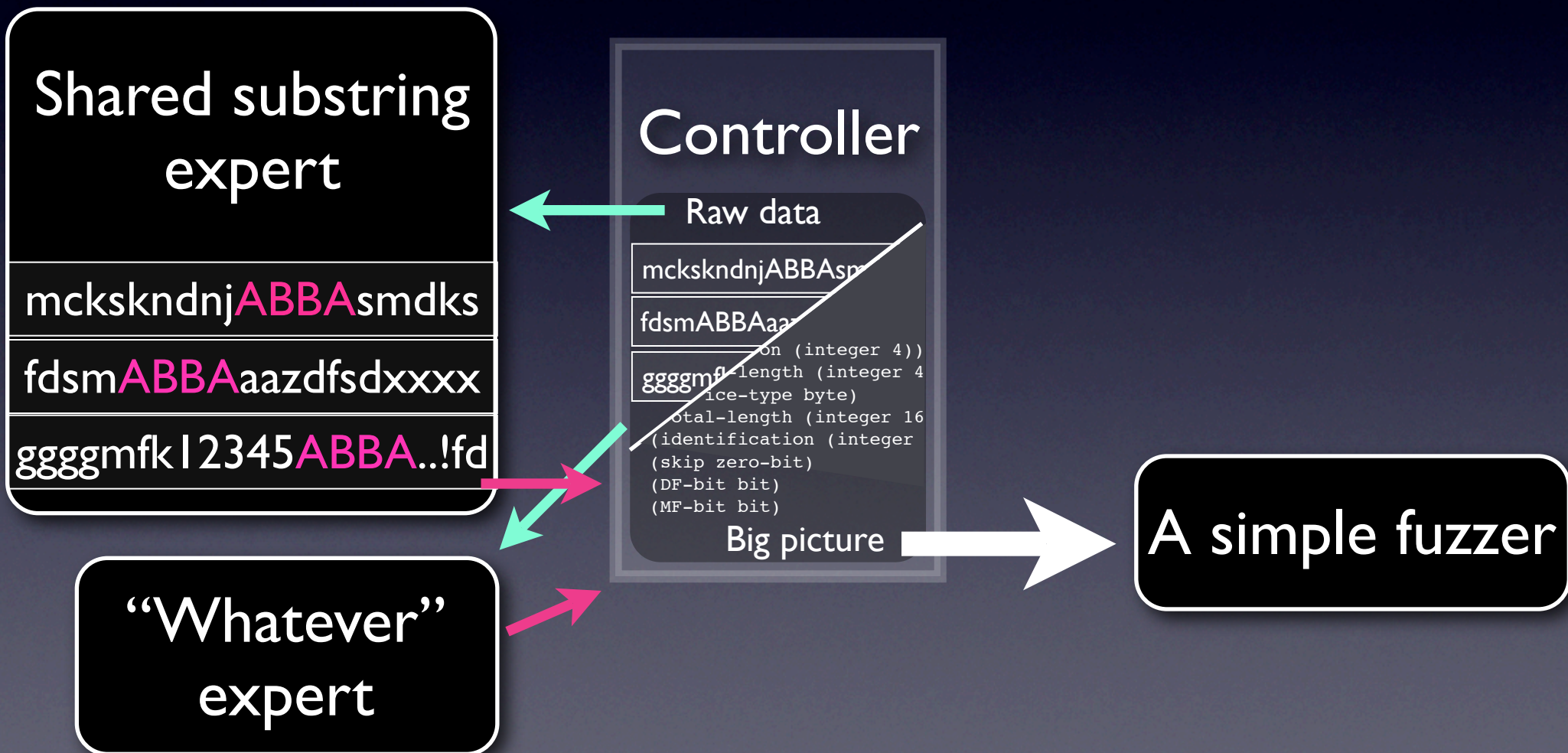
# “Could you **fuzz** with it?”



# “Could you **fuzz** with it?”



# “Could you **fuzz** with it?”



3rd generated file exposed a vulnerability.

\*sob\*

# “Could you **fuzz** with it?”

**Context free grammar!**

Shared substring expert

mckskndnjABBA smdks

fdsmABBAaazdfsdxxxx

ggggmfk12345ABBA..!fd

“Whatever” expert

Controller

Raw data

mckskndnjABBA smdks

fdsmABBAaazdfsdxxxx

ggggmfk12345ABBA..!fd

8888 ice-ice ice (integer 4

total-length (integer 16

(identification (integer

(skip zero-bit)

(DF-bit bit)

(MF-bit bit)

Big picture

A simple fuzzer



# SIMPLER INFERENCE

Seems that context free grammars (CFGs)  
can be pretty powerful.

A context free grammar:

THE GRAMMAR

$0 \rightarrow DE$

$1 \rightarrow 0A$

$2 \rightarrow 2D$

A context free grammar inference algorithm:  
SEQUITUR

# EVEN SIMPLER INFERENCE: MADAM

Replace the most frequent pair of symbols (digram) with a new symbol.

Repeat until all digrams are unique.

# EVEN SIMPLER INFERENCE: MADAM

DEADBEEFDEADABBA

THE GRAMMAR

A diagram consisting of a small black rectangular box at the top containing the text 'THE GRAMMAR'. A white line extends from the right side of this box, curves downwards and then leftwards, forming a larger rounded rectangular frame that is currently empty.

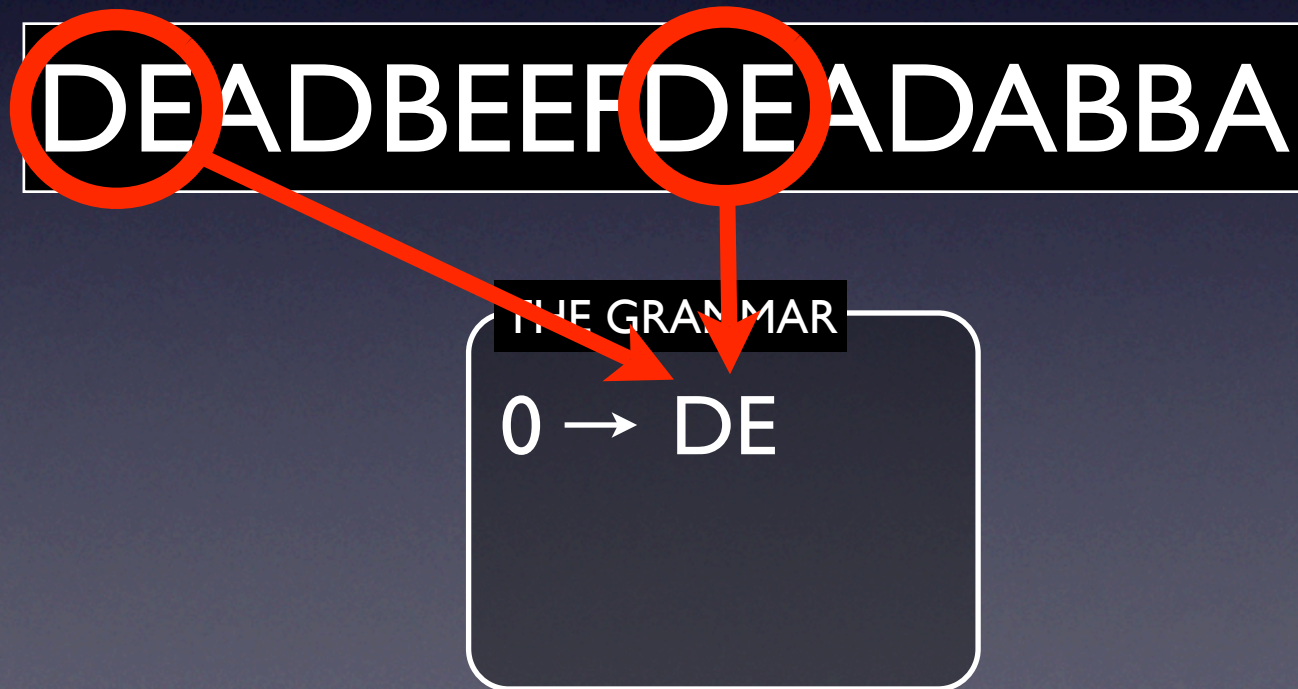
# EVEN SIMPLER INFERENCE: MADAM

DEADBEEFDEADABBA

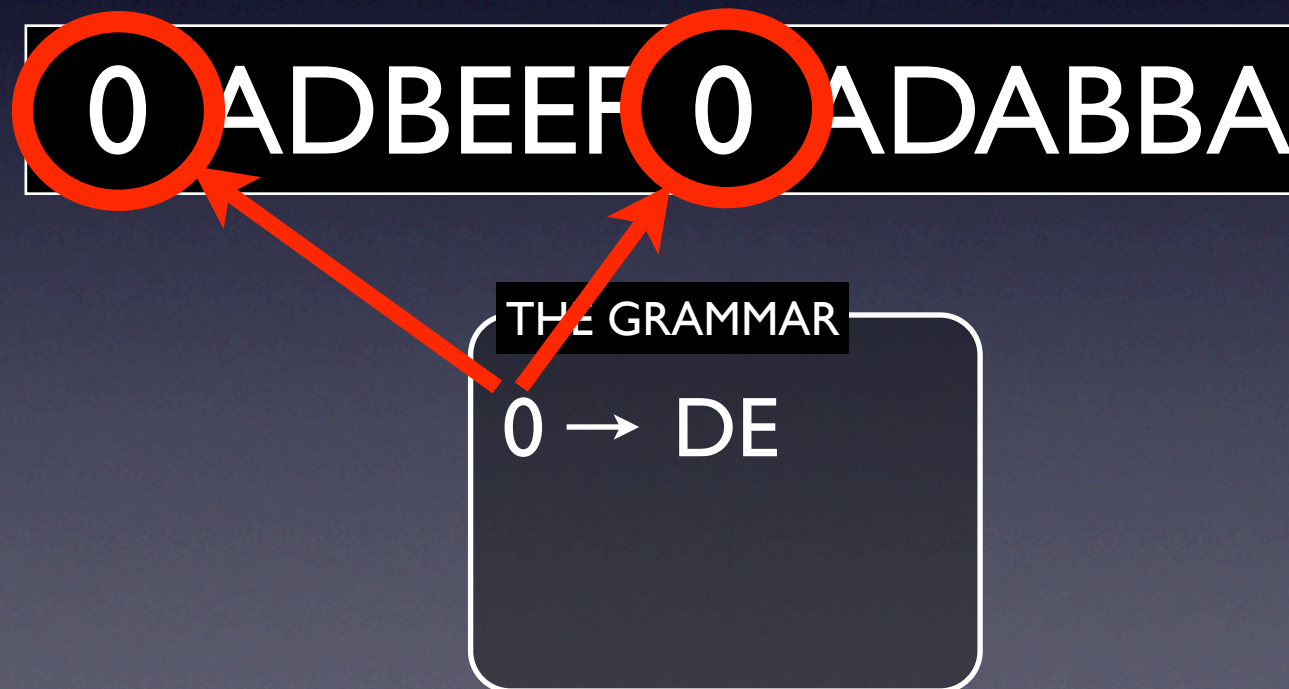
THE GRAMMAR



# EVEN SIMPLER INFERENCE: MADAM



# EVEN SIMPLER INFERENCE: MADAM



# EVEN SIMPLER INFERENCE: MADAM

0 ADBEEF 0 ADABBA

THE GRAMMAR

0 → DE

# EVEN SIMPLER INFERENCE: MADAM

0 ADBEEF 0 ADABBA

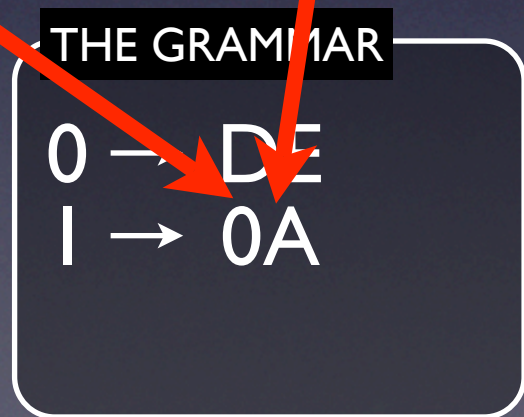
THE GRAMMAR

$0 \rightarrow DE$

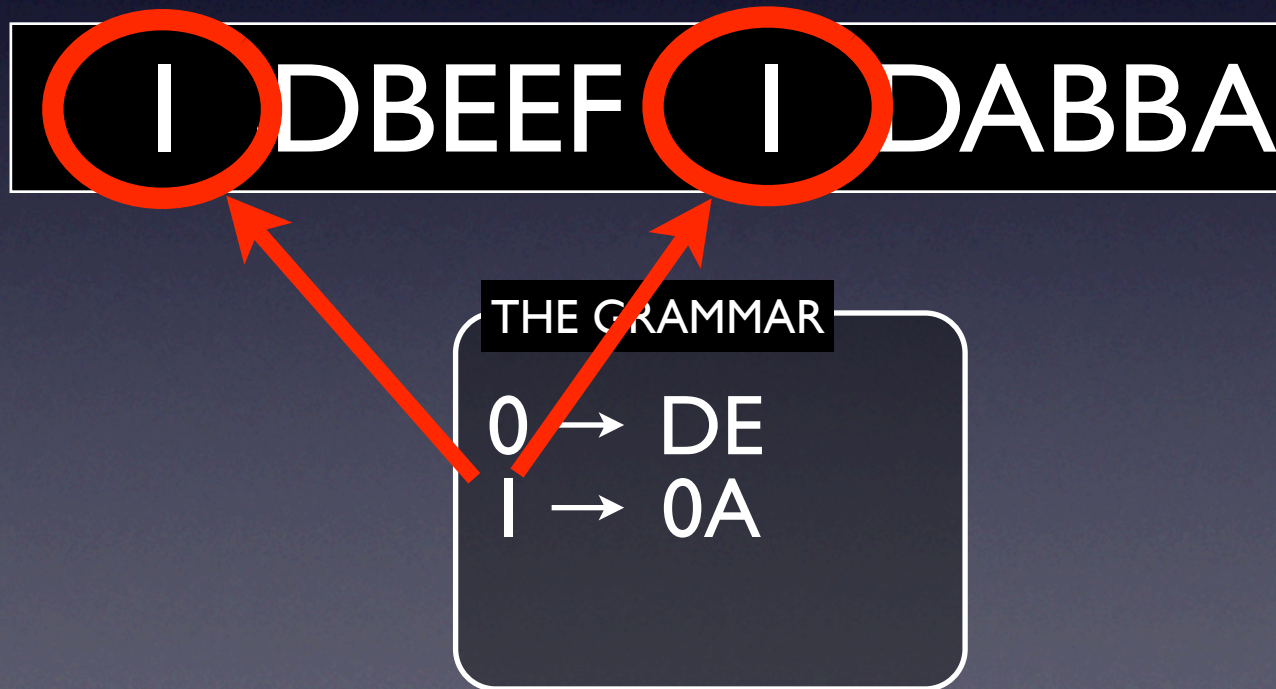


# EVEN SIMPLER INFERENCE: MADAM

0 ADBEEF 0 ADABBA



# EVEN SIMPLER INFERENCE: MADAM



# EVEN SIMPLER INFERENCE: MADAM

I DBEEF I DABBA

THE GRAMMAR

$0 \rightarrow DE$

$I \rightarrow 0A$

# EVEN SIMPLER INFERENCE: MADAM

I DBEEF I DABBA

THE GRAMMAR

$0 \rightarrow DE$

$I \rightarrow 0A$

# EVEN SIMPLER INFERENCE: MADAM

I DBEEF I DABBA

THE GRAMMAR

0 → DE

1 → OA

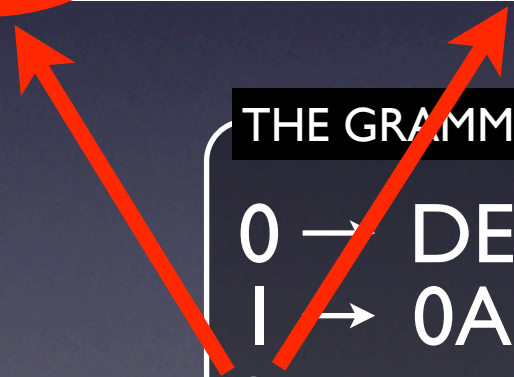
2 → ID

# EVEN SIMPLER INFERENCE: MADAM

**2** BEEF **2** ABBA

THE GRAMMAR

0	→	DE
1	→	0A
2	→	ID



# EVEN SIMPLER INFERENCE: MADAM

2 BEEF 2 ABBA

## THE GRAMMAR

0 → DE

1 → 0A

2 → 1D

# EVEN SIMPLER INFERENCE: MADAM

2 BEEF 2 ABBA

## THE GRAMMAR

0 → DE  
1 → DEA  
2 → ID



# EVEN SIMPLER INFERENCE: MADAM

2 BEEF 2 ABBA

## THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD

# EVEN SIMPLER INFERENCE: MADAM

Can be done to several texts simultaneously.

Can be done in  $O(n)$  time.

Implementation details in “*Offline dictionary-based compression*” by N. J. Larsson and A. Moffat (2000).


# EVEN SIMPLER INFERENCE:

**RE-PAIR**

Can be done to several texts simultaneously.

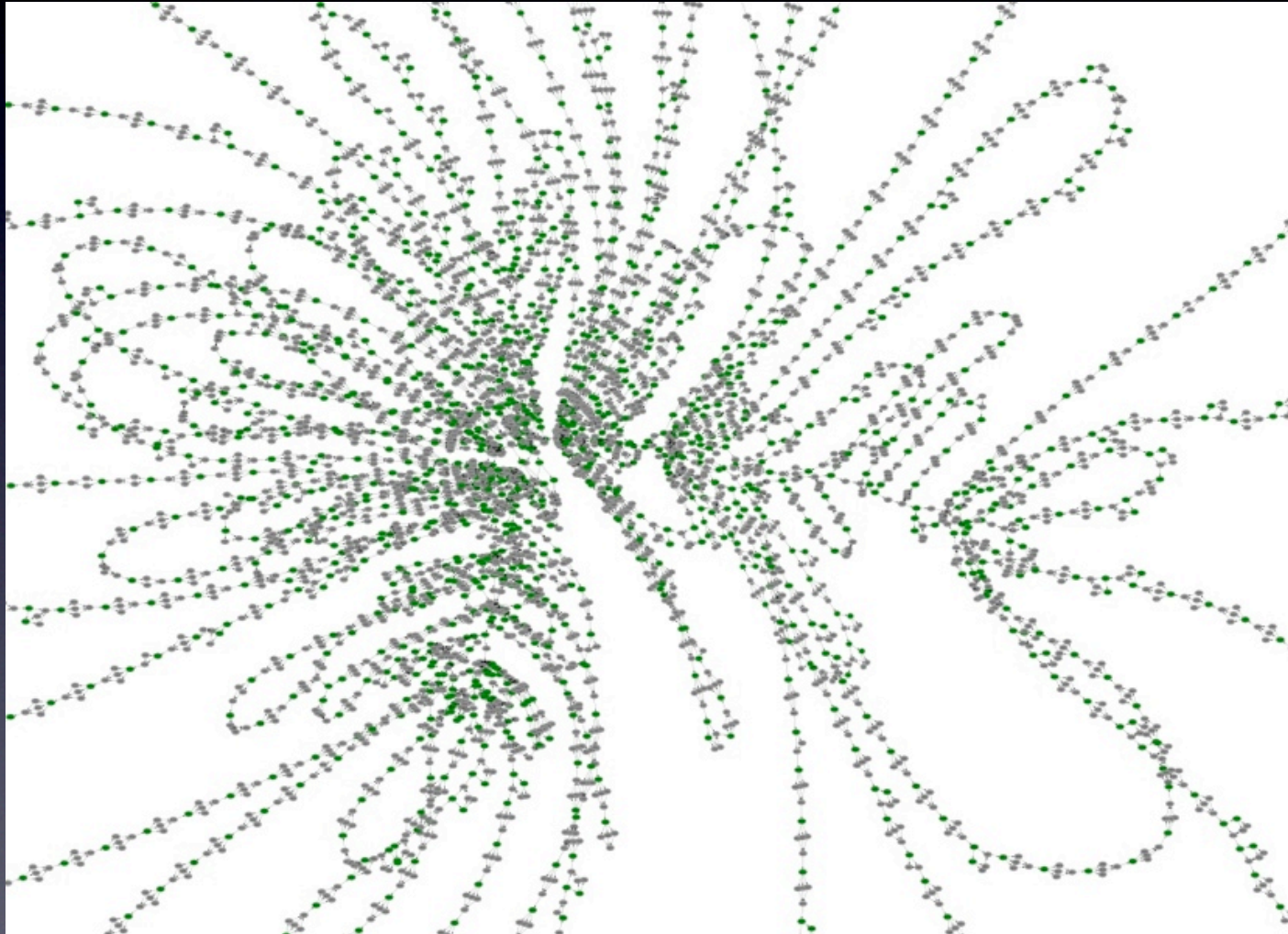
Can be done in  $O(n)$  time.

Implementation details in “*Offline dictionary-based compression*” by N. J. Larsson and A. Moffat (2000).

A man in a grey sweater and tan pants stands on a rooftop, shouting with his arms raised. A large white speech bubble with a black outline points towards him from the right. The background shows a green corrugated metal wall and a cloudy sky.

**Curse you**  
**N. JESPER LARSSOONNN**

# EVEN SIMPLER INFERENCE IN PRACTICE



# FUZZING

2 BEEF 2 ABBA

## THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD

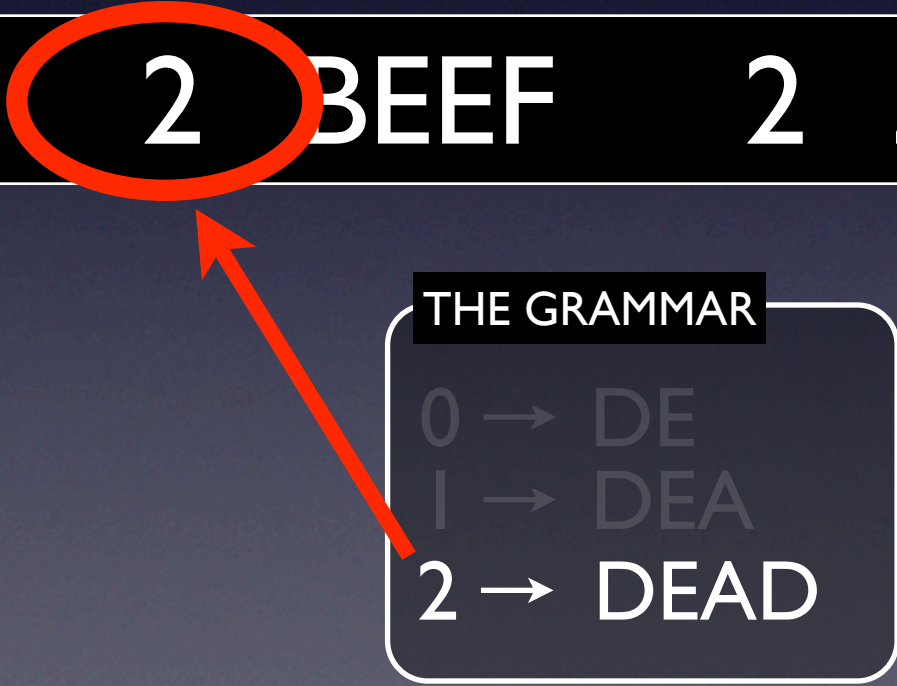
# FUZZING

Normal step

**2** BEEF      2 ABBA

THE GRAMMAR

0 → DE  
1 → DEA  
2 → DEAD



# FUZZING

Normal step

DEADBEEF 2 ABBA

THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD



# FUZZING

Normal step

DEADBEEF    2    ABBA

THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD

# FUZZING

Mutated step

DEADBEEF **2** ABBA

THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD



# FUZZING

Mutated step

DEADBEEF

ABBA

THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD



# FUZZING

Mutated step

DEADBEEF

ABBA

THE GRAMMAR

0 → DE

1 → DEA

2 → DEAD

# RESULTS?

Created all kinds of valid archive files (RAR, BZ2, ...) and inferred grammar for each format.

Fuzzed preliminary test cases, scanned them with AV tools.

## Result summary by archive format

<i>Subject</i>	<i>ace</i>	<i>arj</i>	<i>bz2</i>	<i>cab</i>	<i>gz</i>	<i>lha</i>	<i>rar</i>	<i>tar</i>	<i>zip</i>	<i>zoo</i>
1	x	x	x	x	-	x	-	-	x	x
2	-	x	n/a	x	-	x	x	-	-	n/a
3	-	x	x	x	-	x	x	-	-	-
4	-	x	-	-	-	x	x	-	x	-
5	n/a	n/a	n/a	-	-	n/a	n/a	-	-	n/a

# RESULTS?

A test set with too many (1 632 691) test files.

Coordinated by CERT-FI, JPCERT and CPNI  
(Centre For Protection of National Infrastructure,  
used to be NISCC).

Curse you  
**TAVIS ORMANDYYYYYYYY**



*"[...] break and crash products from at least 40 vendors — including several antivirus vendors... [...]"*

- F-Secure



**huh?**  
“[...] break and crash products from at least 40 vendors — including several antivirus vendors... [...]”

- F-Secure

# 2008! THE FUTURE!

**Plagiarism detector!**

# 2008! THE FUTURE!

Plagiarism detector!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!  
Plagiarism detector!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!  
Plagiarism detector!

More test sets!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!

Plagia

**Metamodeling!**

More test sets!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!

Plagiarism modeling!

Entropy based fuzzing!

More test sets!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!

Plagiarism modeling!

Entropy based fuzzing!

**Network Hoover!**

Back to roots!

Checksum finder!

Length-prefix finder!



# 2008! THE FUTURE!

Protocol paleontology!

Plagiarism modeling!

Entropy based fuzzing!

**Public tools?**

work Hoover!

Back to roots!

Checksum finder!

Length-prefix finder!

# 2008! THE FUTURE!

Protocol paleontology!

Plagiarism modeling!

Entropy based fuzzing!

Put your work Hoover!

**Actual hard data!**

Checksum finder!

Length-prefix finder!

2008! THE FUTURE!



Pro

Entropy

Put

Be

oover!

**QUESTIONS?**

Length-prefix tinder!