Computing Low-Weight DLs

Ryan Henry
Indiana University

Sarah Plosker
Brandon University

Bailey Kacsmar
University of Waterloo

$H((g^B)^A) = \text{🔑}$ $\text{🔑} = H((g^A)^B)$

g^A

g^B

Discrete Exponentiation problem:

an integer (from $\mathbb{Z}_q^*$)

Given (g, x), find g^x

generator of a finite group (of order q)

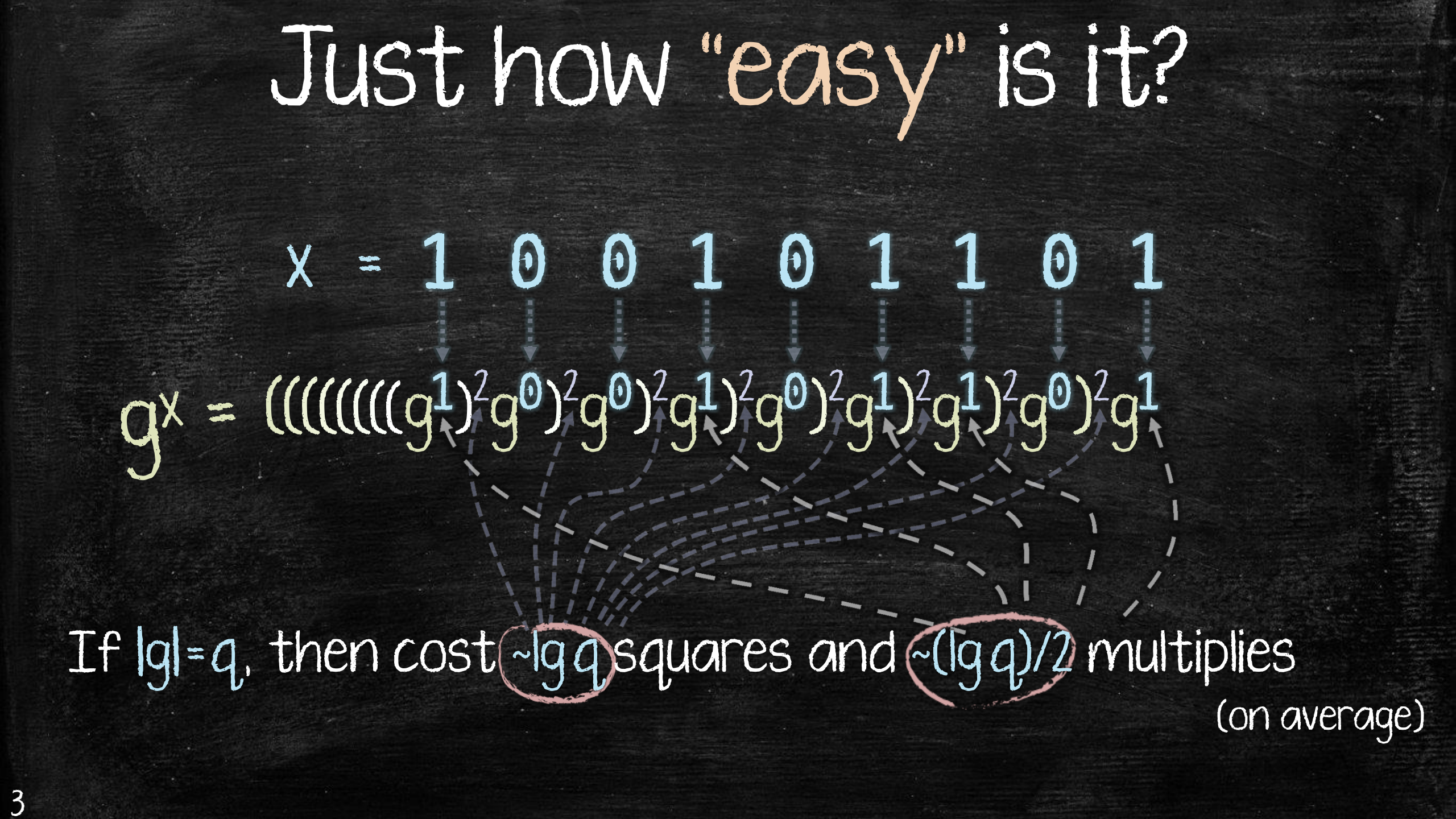
The discrete exponentiation problem is "easy"

(in the sense that there are PPT ALGORITHMS that solve ARBITRARY discrete exponentiation instances)

Just how "easy" is it?

$$x = 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1$$

$$g^x = (((((((g^1)^2 g^0)^2 g^0)^2 g^1)^2 g^0)^2 g^1)^2 g^1)^2 g^0)^2 g^1$$



If $|g| = q$, then cost ~lg q squares and ~(lg q)/2 multiplies (on average)

Discrete Logarithm (DL) problem:

an integer (from $\mathbb{Z}_q^*$)

Given (g, g^x), find x

generator of a finite group (of order q)

Discrete Logarithm (DL) assumption: (hand-wavy version)

∃ infinite families of groups w.r.t which

The DL problem is "hard"

(in the sense that no PPT Algorithm can solve UNIFORM RANDOM DL instances in the groups comprising the family)

Just how "hard" is it?

"Baby-step"

$g, g^2, g^3, \ldots, g^{\lceil\sqrt{q}\rceil-1}$

"Giant-step"

$g^{\lceil\sqrt{q}\rceil}, g^{2\lceil\sqrt{q}\rceil}, \ldots, g^{\lceil\sqrt{q}\rceil^2}$

$$g^x / g^a \stackrel{?}{=} g^{b\lceil\sqrt{q}\rceil}$$

$$\Rightarrow g^x = g^{a+b\lceil\sqrt{q}\rceil}$$

Just how "hard" is it?

"Baby-step"

$g, g^2, g^3, \ldots, g^{\lceil\sqrt{q}\rceil-1}$

"Giant-step"

$g^{\lceil\sqrt{q}\rceil}, g^{2\lceil\sqrt{q}\rceil}, \ldots, g^{\lceil\sqrt{q}\rceil^2}$

If $|g|=q$, then cost $\leq 2q^{1/2}$ multiplies

Cost of "square-and-multiply" grows with number of 1 digits in the exponent

"Hamming weight"

Bright idea (?):

Choose exponents having few 1 bits!

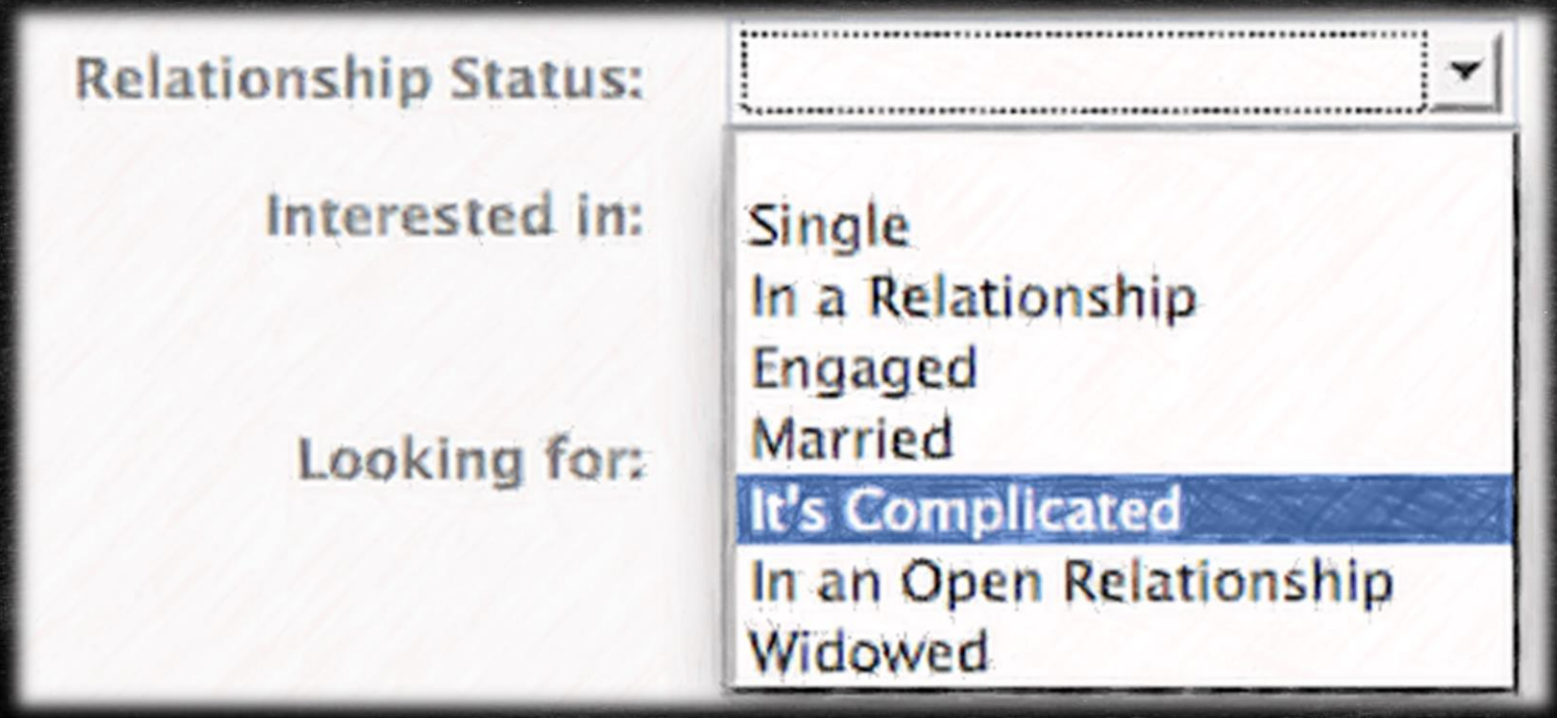
Low-Hamming-Weight DL (LHW-DL) problem:

and a *promise* that "x has Hamming weight $t \ll \lg q$"

Given (g, g^x), find x

Q: How hard is the LHW-DL problem?

The LHW-DL problem is "sorta hard"



How hard is "sorta hard"?

"Baby-step"

$$\forall X_1 \in \binom{[\lg q]}{\lceil t/2 \rceil},\ g^{\sum_{i \in X_1} 2^i}$$

"Giant-step"

$$\forall X_2 \in \binom{[\lg q]}{\lfloor t/2 \rfloor},\ g^x / g^{\sum_{i \in X_2} 2^i}$$

$$g^x / g^{\sum_{i \in X_2} 2^i} \stackrel{?}{=} g^{\sum_{i \in X_1} 2^i}$$

$$\Rightarrow g^x = g^{\sum_{i \in X_1 \cup X_2} 2^i}$$

How hard is "sorta hard"?

"Baby-step"

$$\forall X_1 \in \binom{[\lg q]}{\lceil t/2 \rceil},\ g^{\Sigma_{i \in X_1} 2^i}$$

"Giant-step"

$$\forall X_2 \in \binom{[\lg q]}{\lfloor t/2 \rfloor},\ g^x / g^{\Sigma_{i \in X_2} 2^i}$$

If $|g| = q$, then cost $\sim 2\binom{[\lg q]}{t/2}$ exps

Optimizations

1. Minimal change ordering

   ⇒ *exps in cost become mults!!*

2. Interleaving baby- and giant-steps

   (large constant plus) ⇒ small asymptotic speedup

3. Iterate over "splitting systems"

   ⇒ asymptotic speedup $2^{\binom{\lceil \lg q \rceil}{\lfloor t/2 \rfloor}} \to t^{\binom{\lceil \lg q \rceil / 2}{\lfloor t/2 \rfloor}}$

best known "deterministic" complexity

Coppersmith's algorithm

$x = 1100101011101100000010$

Run baby step over half the bits

Run giant step over other half

No collision? → Shift halves by 1 bit (cyclically)

Coppersmith's algorithm

$x = 1100101010111011000000010$

Run baby step over half the bits

Run giant step over other half

Thm: ∀x, ∃ some shift that yields a collision!

No collision?

(cyclically)

Coppersmith's algorithm

$$x = 11001010111011000010$$

7 ones, 4 zeros

3 ones, 8 zeros

Coppersmith's algorithm

$x = 1100101011101100000010$

~~7 ones, 1 zeros~~

6 ones, 5 zeros

~~3 ones, 6 zeros~~

4 ones, 7 zeros

Coppersmith's algorithm

$x = 11001010111011000010$

~~7 ones, 1 zeros~~

~~6 ones, 5 zeros~~

~~6 ones, 1 zeros~~

~~7 ones, 2 zeros~~

~~7 ones, 1 zeros~~

~~6 ones, 2 zeros~~

~~6 ones, 5 zeros~~

5 ones, 6 zeros

match found!!

~~3 ones, 4 zeros~~

~~4 ones, 7 zeros~~

~~4 ones, 1 zeros~~

~~3 ones, 3 zeros~~

~~3 ones, 1 zeros~~

~~4 ones, 2 zeros~~

~~4 ones, 7 zeros~~

5 ones, 6 zeros

Coppersmith's algorithm

$x = 1100101011101100000010$

- Two loops:
  - "Outer loop" runs over $m/2$ cyclic shifts
  - "Inner loop" iterates over $\leq 2 \times \binom{\lceil (\lg q)/2 \rceil}{\lfloor t/2 \rfloor}$

$\Rightarrow$ Total cost: $\leq m \binom{\lceil (\lg q)/2 \rceil}{\lfloor t/2 \rfloor}$

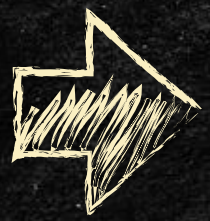
Pascal's Lemma

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

$\binom{n-1}{k}$ of the $\binom{n}{k}$ values in each iteration were also computed in the previous iteration!

Pascal's Lemma

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$



Can save a factor $\approx \binom{\lceil (\lg q)/2 \rceil}{\lfloor t/2 \rfloor} / \binom{\lceil (\lg q)/2 \rceil - 1}{\lfloor t/2 \rfloor - 1} \approx m/t$ work

Pascal's Lemma

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Total cost: $\leq t\binom{\lceil (\lg q)/2 \rceil}{\lfloor t/2 \rfloor} + o(1)$

Low-Radix-b-Weight DL (LRW_b-DL) problem:

and a *promise* that
"x has Radix-b weight $t \ll \log_b q$"

Given (g, g^x): find x

Q: How hard is the Low-Radix-b-Weight DL problem?

The LRW$_b$-DL problem is about as hard as the LHW-DL problem

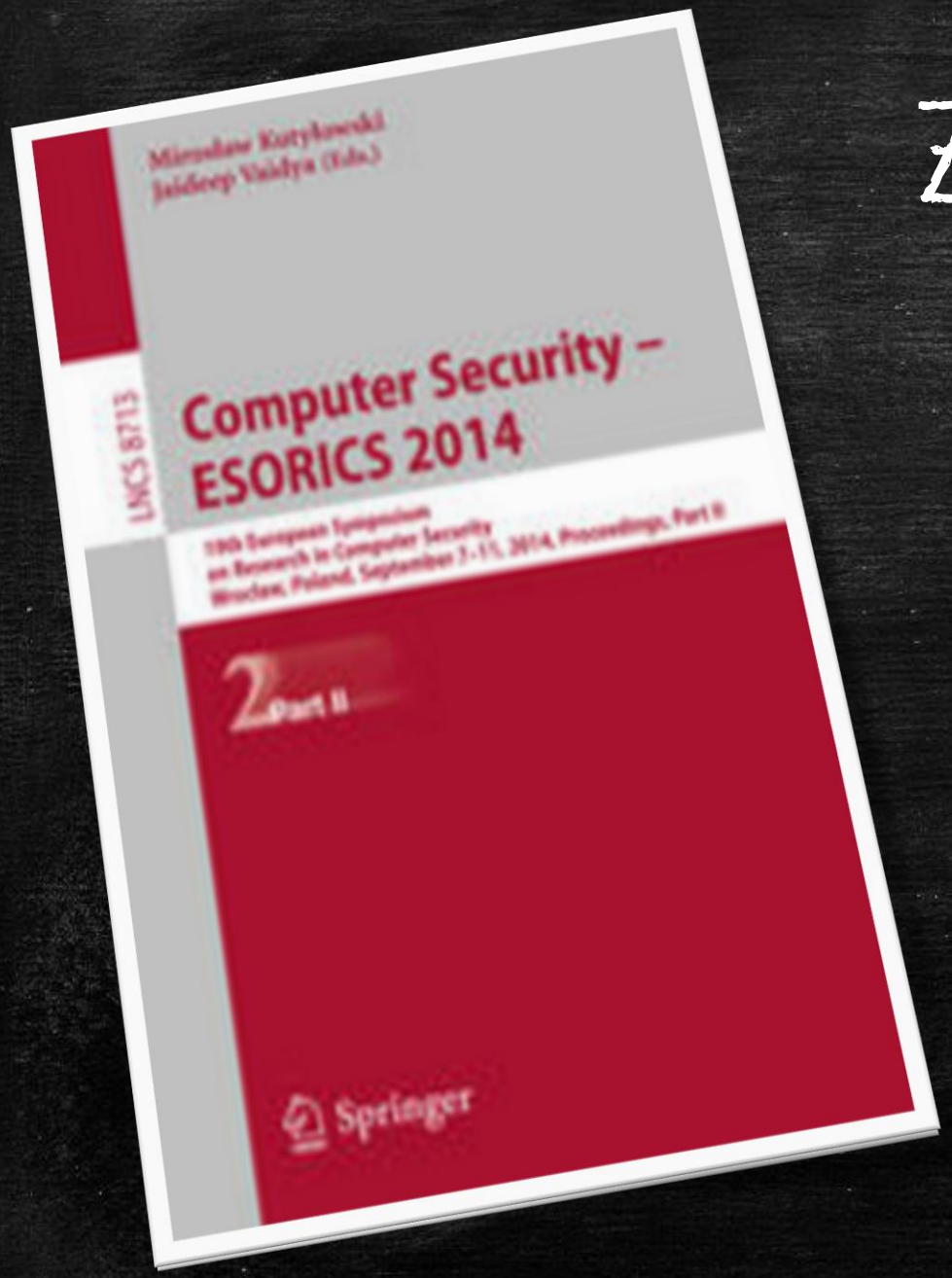
Add "innermost loop" over the $(b-1)^{t/2}$ possibilities for the non-zero digits

⇒ pick up an extra $(b-1)^{t/2}$ factor in cost

- partially offset by shorter radix-b length and (if we're lucky) lower radix-b weight

THM: (hand-wavy version)

If $B > b$ and radix-B density $\leq$ radix-b density, then radix-B algorithm is faster



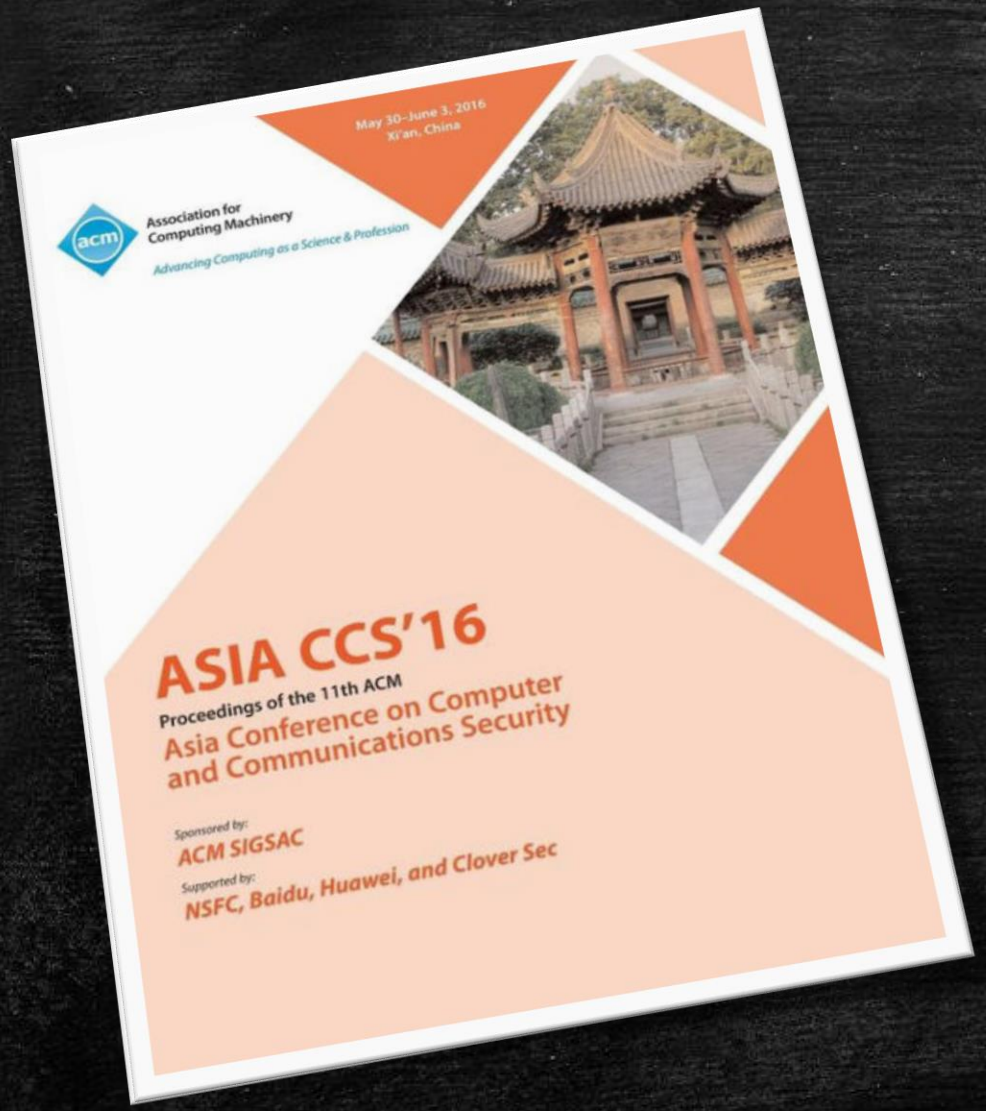
Zero-knowledge Password Policy Checks and Verifier-based PAKE

Kiefer and Manulis

ESORICS 2014

Before: "Provably secure"

Now: "Demonstrably insecure"



Blind Password Registration for Verifier-Based PAKE

Kiefer and Manulis

AsiaPKC 2016

Before: "Provably secure"

Now: "Demonstrably insecure"

A Provably-Secure and Efficient Verifier-Based Anonymous Password-Authenticated Key Exchange Protocol

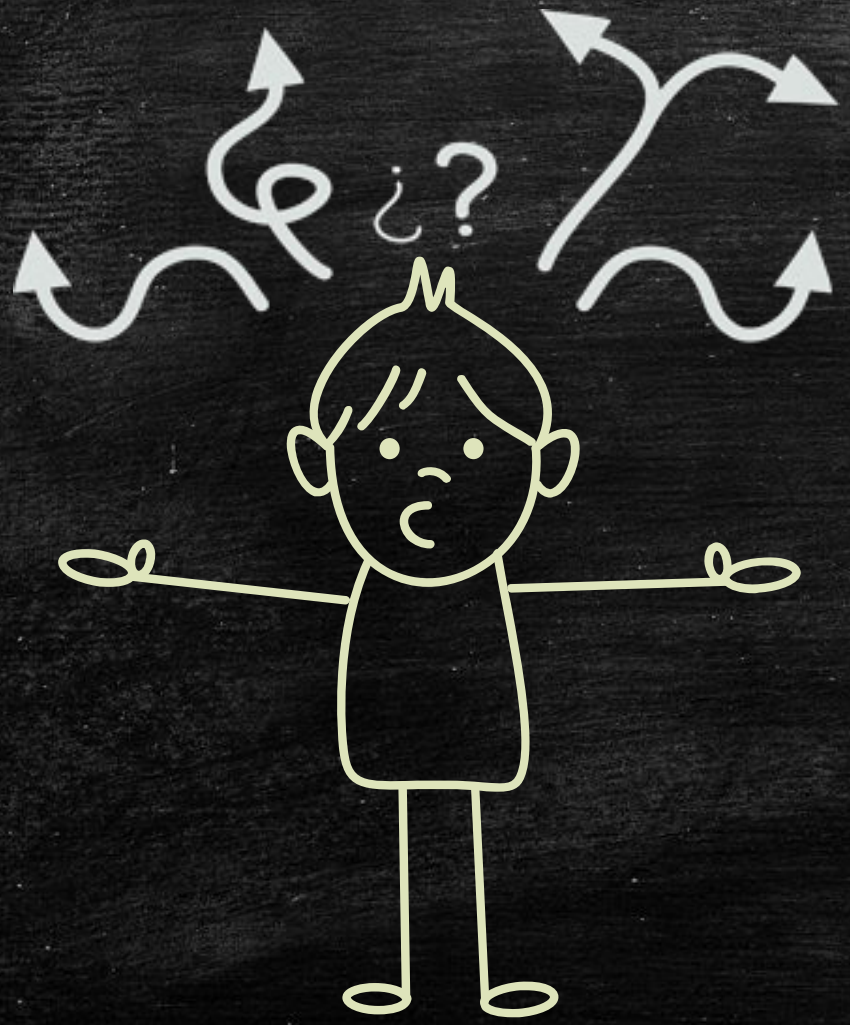
Yang, Jiang, Xu, Hou, Zhao, and Choo

TrustCom/BigDataSE/ISPA 2016

Before: "Provably secure"

Now: "Demonstrably insecure"

Where do we go from here?



Lattice crypto!

- Low-weight secret keys (vectors)
- low weight linear combinations
- other risky "low-weight" ideas...

That's all for today, folks!