ECC 2012, 10/30/2012

QUERETARO, MEXICO

Non-uniformity

Neal Koblitz, Univ. of Washington, koblitz@uw.edu

Based on joint work with Alfred Menezes – see http://anotherlook.ca especially "Another look at HMAC" and "Another look at Non-uniformity."

Based on joint work with Alfred Menezes – see http://anotherlook.ca especially "Another look at HMAC" and "Another look at Non-uniformity."

Outline:

1. Story of recent dispute about non-uniformity – including some words of Clint Eastwood.

Based on joint work with Alfred Menezes – see http://anotherlook.ca especially "Another look at HMAC" and "Another look at Non-uniformity."

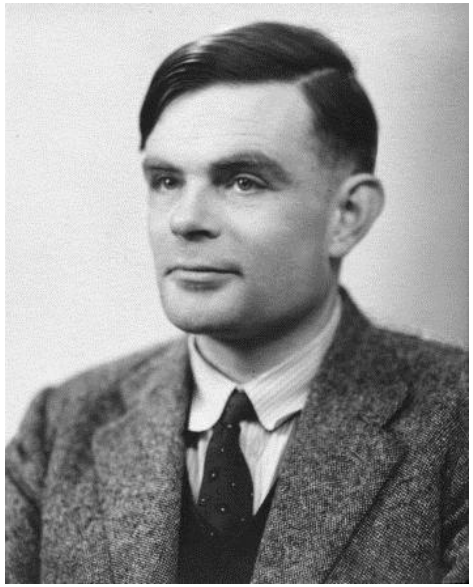
Outline:

1. Story of recent dispute about non-uniformity – including some words of Clint Eastwood.

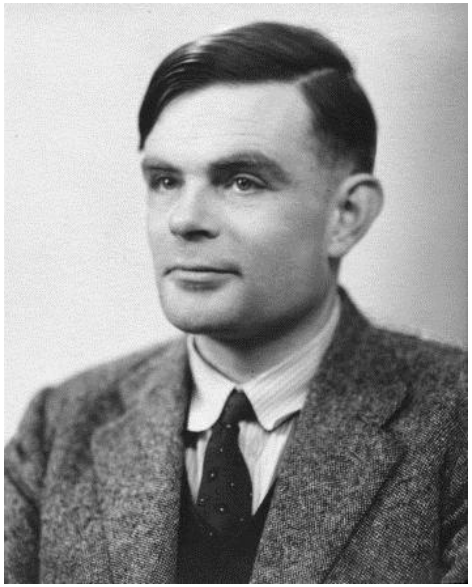
2. How this controversy ties in with ECF & ECDLP.

Before beginning I'd like to pay tribute to two great mathematicians:

Before beginning I'd like to pay tribute to two great mathematicians:

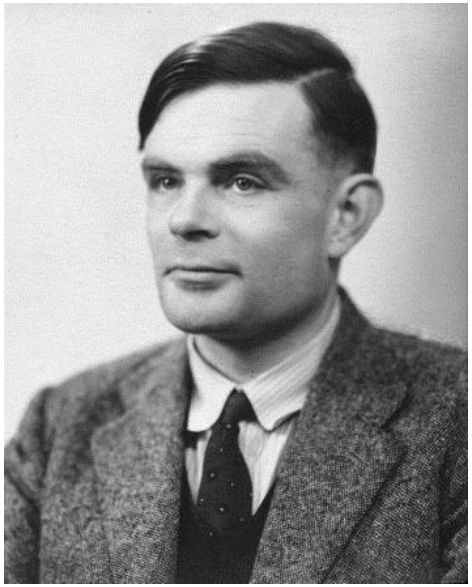


Before beginning I'd like to pay tribute to two great mathematicians:



Alan Turing (1912-1954) invented the modern notion of a uniform algorithm that is central to any discussion of computational complexity.

Before beginning I'd like to pay tribute to two great mathematicians:



Alan Turing (1912-1954) invented the modern notion of a uniform algorithm that is central to any discussion of computational complexity.

Our work on HMAC and non-uniformity can be viewed as a vindication of Turing's viewpoint in the setting of provable security in cryptography.

Mark Kac (1914-1984) did fundamental work in probabilities, statistical mechanics, and other applications.

Mark Kac (1914-1984) did fundamental work in probabilities, statistical mechanics, and other applications. He wrote one of the best expository mathematics books ever: *Statistical Independence in Probability, Analysis and Number Theory* (1969).

Let f be a compression function

$f: \{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$, $b > c$.

Let f be a compression function

$f: \{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$, b > c.

Here typically b=512 and
c=160 (SHA1) or c=128 (MD5).

Let f be a compression function

$f: \{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$, $b > c$.

Here typically b=512 and c=160 (SHA1) or c=128 (MD5).

It is used to construct the hash-based message authentication code HMAC.

Bellare-Canetti-Krawczyk (1996) gave a security proof for HMAC. They proved that HMAC has the secure-MAC property – which means message tags are hard to forge even under chosen-message attack.

Bellare-Canetti-Krawczyk (1996) gave a security proof for HMAC. They proved that HMAC has the secure-MAC property – which means message tags are hard to forge even under chosen-message attack.

They made two assumptions: (1) that the underlying compression function f has the secure-MAC property

Bellare-Canetti-Krawczyk (1996) gave a security proof for HMAC. They proved that HMAC has the secure-MAC property – which means message tags are hard to forge even under chosen-message attack.

They made two assumptions: (1) that the underlying compression function f has the secure-MAC property, and (2) that the corresponding iterated hash function has a certain collision-resistance property.

Bellare-Canetti-Krawczyk (1996) gave a security proof for HMAC. They proved that HMAC has the secure-MAC property – which means message tags are hard to forge even under chosen-message attack.

They made two assumptions: (1) that the underlying compression function f has the secure-MAC property, and (2) that the corresponding iterated hash function has a certain collision-resistance property.

After Wang (2005) showed how to find collisions for MD5 (and similar results were likely for SHA1), the 1996 proof lost its usefulness as a real-world security guarantee.

At Crypto 2006 Bellare gave a new proof of HMAC security without assuming collision-resistance, but rather assuming that the compression function f is a secure pseudorandom function (prf).

At Crypto 2006 Bellare gave a new proof of HMAC security without assuming collision-resistance, but rather assuming that the compression function f is a secure pseudorandom function (prf).

Bellare's main theorem:

If f is a prf, then so is HMAC.

At Crypto 2006 Bellare gave a new proof of HMAC security without assuming collision-resistance, but rather assuming that the compression function f is a secure pseudorandom function (prf).

Bellare's main theorem:

If f is a prf, then so is HMAC.

Bellare claimed that his theorem justifies HMAC "up to roughly $2^{c/2}/n$ queries," where n is the number of message blocks allowed, say $n = 2^{20}$.

At Crypto 2006 Bellare gave a new proof of HMAC security without assuming collision-resistance, but rather assuming that the compression function f is a secure pseudorandom function (prf).

Bellare's main theorem:

If f is a prf, then so is HMAC.

Bellare claimed that his theorem justifies HMAC "up to roughly $2^{c/2}/n$ queries," where n is the number of message blocks allowed, say $n = 2^{20}$. This means up to 2^{44} queries for HMAC-MD5, and up to 2^{60} queries for HMAC-SHA1.

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$.

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$. We write T=f(K,M), where T is the c-bit tag determined by a c-bit key K and a b-bit message M.

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$. We write T=f(K,M), where T is the c-bit tag determined by a c-bit key K and a b-bit message M.

The adversary A is presented with an oracle O which gives a c-bit output in response to message-queries of A.

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$. We write T=f(K,M), where T is the c-bit tag determined by a c-bit key K and a b-bit message M.

The adversary A is presented with an oracle O which gives a c-bit output in response to message-queries of A. The oracle with equal probability is either (1) f(K,.) for some random key K, or else (2) a random function.

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$. We write T=f(K,M), where T is the c-bit tag determined by a c-bit key K and a b-bit message M.

The adversary A is presented with an oracle O which gives a c-bit output in response to message-queries of A. The oracle with equal probability is either (1) f(K,.) for some random key K, or else (2) a random function.

After q queries, A must guess whether O is (1) or (2) with significantly more than 50% accuracy

Definition of prf-security.

Recall f: $\{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$. We write T=f(K,M), where T is the c-bit tag determined by a c-bit key K and a b-bit message M.

The adversary A is presented with an oracle O which gives a c-bit output in response to message-queries of A. The oracle with equal probability is either (1) f(K,.) for some random key K, or else (2) a random function.

After q queries, A must guess whether O is (1) or (2) with significantly more than 50% accuracy, i.e., with probability ½ + Adv(A), where Adv(A) is the "advantage of A."

In January 2012, Alfred and I were thinking about HMAC and were becoming increasingly bothered by a certain crucial step in Bellare's proof.

In January 2012, Alfred and I were thinking about HMAC and were becoming increasingly bothered by a certain crucial step in Bellare's proof.

Bellare's proof consists of several lemmas and "game-hopping" diagrams in 12 very hard-to-read pages.

In January 2012, Alfred and I were thinking about HMAC and were becoming increasingly bothered by a certain crucial step in Bellare's proof.

Bellare's proof consists of several lemmas and "game-hopping" diagrams in 12 very hard-to-read pages.

The following summary is informal and much simplified, but captures essentially what is going on.

In January 2012, Alfred and I were thinking about HMAC and were becoming increasingly bothered by a certain crucial step in Bellare's proof.

Bellare's proof consists of several lemmas and "game-hopping" diagrams in 12 very hard-to-read pages.

The following summary is informal and much simplified, but captures essentially what is going on.

At a key point in the proof Bellare describes an adversary of the prf-property of the compression function f that arises when HMAC fails to have the desired security property.

He sets (M*,M´*) equal to a pair of b-bit messages for which the probability of a collision f(K,M*) = f(K,M´*) (where the probability is taken over all keys K in $\{0,1\}^c$) is maximal.

He sets (M*,M´*) equal to a pair of b-bit messages for which the probability of a collision f(K,M*) = f(K,M´*) (where the probability is taken over all keys K in $\{0,1\}^c$) is maximal.

Using what he calls a "coin-fixing" argument, he hardwires this pair of collision-prone messages into the adversary A

He sets (M*,M´*) equal to a pair of b-bit messages for which the probability of a collision f(K,M*) = f(K,M´*) (where the probability is taken over all keys K in $\{0,1\}^c$) is maximal.

Using what he calls a "coin-fixing" argument, he hardwires this pair of collision-prone messages into the adversary A, which will later query the two messages to the oracle O in order to defeat the prf-test for f.

He sets (M*,M´*) equal to a pair of b-bit messages for which the probability of a collision f(K,M*) = f(K,M´*) (where the probability is taken over all keys K in $\{0,1\}^c$) is maximal.

Using what he calls a "coin-fixing" argument, he hardwires this pair of collision-prone messages into the adversary A, which will later query the two messages to the oracle O in order to defeat the prf-test for f.

This algorithm A exists in the mathematical sense, because the pair (M*,M´*) obviously exists.

He sets (M*,M´*) equal to a pair of b-bit messages for which the probability of a collision f(K,M*) = f(K,M´*) (where the probability is taken over all keys K in $\{0,1\}^c$) is maximal.

Using what he calls a "coin-fixing" argument, he hardwires this pair of collision-prone messages into the adversary A, which will later query the two messages to the oracle O in order to defeat the prf-test for f.

This algorithm A exists in the mathematical sense, because the pair (M*,M´*) obviously exists. But it is unconstructible in the sense that there is no feasible way to construct it.

Our reactions:

Our reactions:

(1) Huh?

Our reactions:

(1) Huh?

(2) Algorithms are a computational notion. What does it mean in practice for an algorithm to exist mathematically if it's unconstructible?

Our reactions:

(1) Huh?

(2) Algorithms are a computational notion. What does it mean in practice for an algorithm to exist mathematically if it's unconstructible?

(3) How can one conclude anything at all about the real-world security of HMAC if one has to assume that the compression function resists even unconstructible adversaries?

Mihir Bellare is the most cited researcher in all of cryptography – 27,551 citations at last count on googlescholar.com.

Mihir Bellare is the most cited researcher in all of cryptography – 27,551 citations at last count on googlescholar.com.

Bellare is also coinventor (along with Rogaway) of the paradigm "practice-oriented provable security," and the introduction to his paper stresses the real-world applicability of results on the security of HMAC, which is deployed far and wide.

Mihir Bellare is the most cited researcher in all of cryptography – 27,551 citations at last count on googlescholar.com.

Bellare is also coinventor (along with Rogaway) of the paradigm "practice-oriented provable security," and the introduction to his paper stresses the real-world applicability of results on the security of HMAC, which is deployed far and wide. It is surprising that someone like that would prove a theorem by means of what we believed to be a bogus argument.

In February we posted a first version of "Another look at HMAC," which explained the flaw in Bellare's proof.

In February we posted a first version of "Another look at HMAC," which explained the flaw in Bellare's proof.

Within hours of the paper's posting on eprint, we received an angry email from Bellare objecting to our use of the word "flaw" and saying that his proof is correct in the non-uniform model of complexity.

In response to my question about why his paper made no mention of the fact that the main theorem was intended to be understood in the non-uniform model and is invalid in the uniform model of complexity, Bellare wrote:

In response to my question about why his paper made no mention of the fact that the main theorem was intended to be understood in the non-uniform model and is invalid in the uniform model of complexity, Bellare wrote:

“My paper uses a concrete complexity framework. Such a framework is inherently non-uniform.

In response to my question about why his paper made no mention of the fact that the main theorem was intended to be understood in the non-uniform model and is invalid in the uniform model of complexity, Bellare wrote:

“My paper uses a concrete complexity framework. Such a framework is inherently non-uniform. This has been understood since such frameworks started….

In response to my question about why his paper made no mention of the fact that the main theorem was intended to be understood in the non-uniform model and is invalid in the uniform model of complexity, Bellare wrote:

“My paper uses a concrete complexity framework. Such a framework is inherently non-uniform. This has been understood since such frameworks started…. it never occurred to me that a reader would not understand that when complexity is concrete, we have non-uniformity.”

In response to my question about why his paper made no mention of the fact that the main theorem was intended to be understood in the non-uniform model and is invalid in the uniform model of complexity, Bellare wrote:

"My paper uses a concrete complexity framework. Such a framework is inherently non-uniform. This has been understood since such frameworks started…. it never occurred to me that a reader would not understand that when complexity is concrete, we have non-uniformity."

A few other leading theoretical cryptographers were also angry at us, and supported Bellare.

From Yehuda Lindell's Discussion Forum posting on eprint: "There is no flaw whatsoever in the HMAC proof.

From Yehuda Lindell's Discussion Forum posting on eprint: "There is no flaw whatsoever in the HMAC proof. The so-called flaw pointed out by Koblitz and Menezes is a standard proof in the non-uniform model (where adversaries are modeled as families of polynomial-size circuits or equivalently as polynomial-time Turing machines with advice).

From Yehuda Lindell's Discussion Forum posting on eprint: "There is no flaw whatsoever in the HMAC proof. The so-called flaw pointed out by Koblitz and Menezes is a standard proof in the non-uniform model (where adversaries are modeled as families of polynomial-size circuits or equivalently as polynomial-time Turing machines with advice). This type of proof is known to anyone who has taken a basic theory of cryptography (or complexity) course…"

From Yehuda Lindell's Discussion Forum posting on eprint: "There is no flaw whatsoever in the HMAC proof. The so-called flaw pointed out by Koblitz and Menezes is a standard proof in the non-uniform model (where adversaries are modeled as families of polynomial-size circuits or equivalently as polynomial-time Turing machines with advice). This type of proof is known to anyone who has taken a basic theory of cryptography (or complexity) course…"

From Jonathan Katz's blog (February 28, 2012):

From Yehuda Lindell's Discussion Forum posting on eprint: "There is no flaw whatsoever in the HMAC proof. The so-called flaw pointed out by Koblitz and Menezes is a standard proof in the non-uniform model (where adversaries are modeled as families of polynomial-size circuits or equivalently as polynomial-time Turing machines with advice). This type of proof is known to anyone who has taken a basic theory of cryptography (or complexity) course…"

From Jonathan Katz's blog (February 28, 2012): "Many researchers are justifiably concerned about the fact that Alfred Menezes will be giving an invited talk at Eurocrypt 2012…. I share this concern."

So the coin-fixing step was not a momentary lapse by a prominent scholar.

So the coin-fixing step was not a momentary lapse by a prominent scholar. It was a permitted step in the "non-uniform model."

So the coin-fixing step was not a momentary lapse by a prominent scholar. It was a permitted step in the "non-uniform model." Bellare was defending his use of unconstructible adversaries by saying that his entire paper was written in the non-uniform model of complexity.

So the coin-fixing step was not a momentary lapse by a prominent scholar. It was a permitted step in the "non-uniform model." Bellare was defending his use of unconstructible adversaries by saying that his entire paper was written in the non-uniform model of complexity.

At this point we reexamined other parts of Bellare's paper – especially his method for concluding that his main theorem justifies HMAC up to $2^{c/2}/n$ queries – with non-uniformity in our minds.

Bellare's main theorem gives concrete bounds (this is what makes it "practice-oriented" provable security) that in principle allow the practitioner to determine how many adversarial queries HMAC can endure before the theorem has no content

Bellare's main theorem gives concrete bounds (this is what makes it "practice-oriented" provable security) that in principle allow the practitioner to determine how many adversarial queries HMAC can endure before the theorem has no content (i.e., before it says nothing at all about HMAC security – this occurs when the probability guarantee becomes >1).

Bellare's main theorem gives concrete bounds (this is what makes it "practice-oriented" provable security) that in principle allow the practitioner to determine how many adversarial queries HMAC can endure before the theorem has no content (i.e., before it says nothing at all about HMAC security – this occurs when the probability guarantee becomes >1).

Thus, if you plug in the best attacks known on the pseudorandomness of the compression functions used in real-world deployments of HMAC – and it's reasonable to assume that these compression functions were designed well enough so that only generic, general-purpose attacks are available – you can solve for the bounds on the parameters for which the theorem loses content.

This is what Bellare does in the section of his Crypto 2006 paper devoted to interpreting his theorem.

This is what Bellare does in the section of his Crypto 2006 paper devoted to interpreting his theorem.

He assumes that the best attack known on prf-ness of f is exhaustive key search. The adversary in that case has advantage of order only 2^{-c}.

This is what Bellare does in the section of his Crypto 2006 paper devoted to interpreting his theorem.

He assumes that the best attack known on prf-ness of f is exhaustive key search. The adversary in that case has advantage of order only 2^{-c}.

In the classical meaning – Turing's meaning – of an algorithm, this assumption is very reasonable, because no one has any idea of a faster uniform algorithm than exhaustive search for this purpose.

This is what Bellare does in the section of his Crypto 2006 paper devoted to interpreting his theorem.

He assumes that the best attack known on prf-ness of f is exhaustive key search. The adversary in that case has advantage of order only 2^{-c}.

In the classical meaning – Turing's meaning – of an algorithm, this assumption is very reasonable, because no one has any idea of a faster uniform algorithm than exhaustive search for this purpose.

But recall that Bellare insisted that he is working in the non-uniform model of complexity, and even claimed that any concrete study of security should be in that model (and that it had never occurred to him that any reader would fail to understand this).

We soon saw that Bellare's assumption is blatantly false in the non-uniform model of complexity.

We soon saw that Bellare's assumption is blatantly false in the non-uniform model of complexity.

In fact, it turns out that it is easy to see why.

We soon saw that Bellare's assumption is blatantly false in the non-uniform model of complexity.

In fact, it turns out that it is easy to see why.

One can describe a very simple but very powerful unconstructible adversary of the prf-ness of f that causes the number of security bits in Bellare's guarantee to drop by more than half – from 60 to 28 for SHA1 and from 44 to 20 for MD5.

We soon saw that Bellare's assumption is blatantly false in the non-uniform model of complexity.

In fact, it turns out that it is easy to see why.

One can describe a very simple but very powerful unconstructible adversary of the prf-ness of f that causes the number of security bits in Bellare's guarantee to drop by more than half – from 60 to 28 for SHA1 and from 44 to 20 for MD5.

To explain this, I will enlist the help of the football monkey.

The football monkey is a cousin of the proverbial monkey-at-a-typewriter, who, given enough time, can randomly type a flawless *Macbeth,* but who in practice is unlikely to get to the end of "Enter three witches…"

The football monkey does not share his cousin's interest in typing flawless copies of great literature.

The football monkey does not share his cousin's interest in typing flawless copies of great literature.

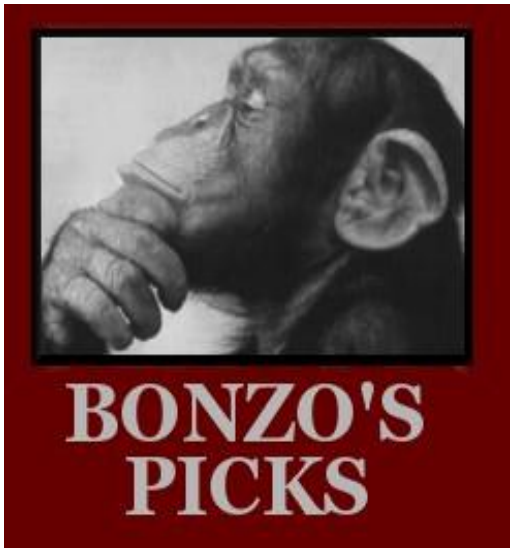
Rather, every week during the American football season he announces his "picks" for the week – which teams he predicts will win.

The football monkey does not share his cousin's interest in typing flawless copies of great literature.

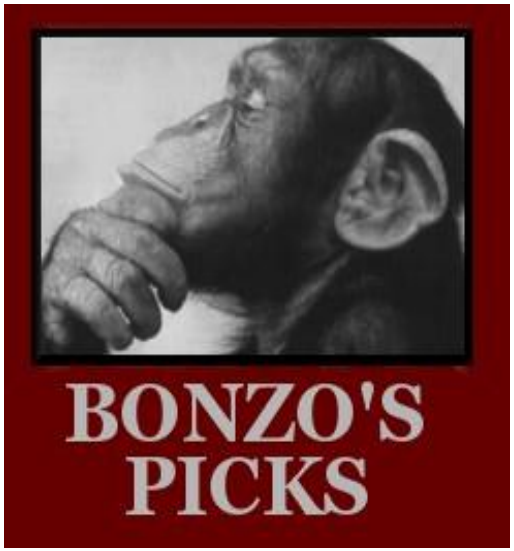
Rather, every week during the American football season he announces his "picks" for the week – which teams he predicts will win.

Much like human football pundits, the football monkey is very proud and pleased with himself whenever he does better than 50% / 50%.

The football monkey's non-uniform attack on NFL pseudorandomness:

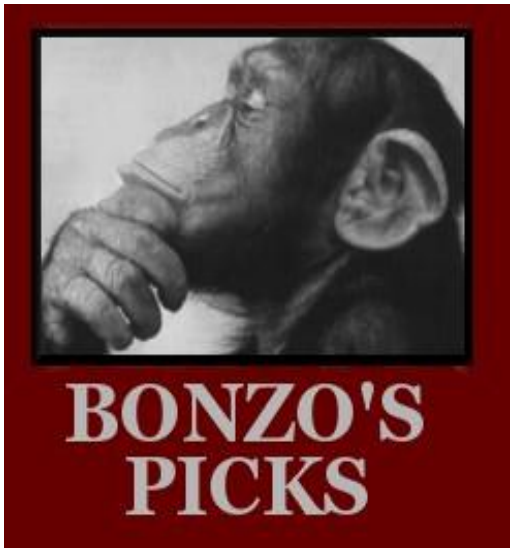


The football monkey's non-uniform attack on NFL pseudorandomness:



Suppose the season has 256 games (not quite true).

The football monkey's non-uniform attack on NFL pseudorandomness:



Suppose the season has 256 games (not quite true). Under this attack the monkey will get at least 144/112 right/wrong rather than 128/128.

Here’s the non-uniform attack:

Here’s the non-uniform attack:

PICK THE MONKEY’S BEST SEASON!

Here's the non-uniform attack:

PICK THE MONKEY'S BEST SEASON!

The theory of random walks says that the standard deviation from 128/128 will be $2^{c/2}$ where $2^c = 256$, i.e., 16.

Here's the non-uniform attack:

PICK THE MONKEY'S BEST SEASON!

The theory of random walks says that the standard deviation from 128/128 will be $2^{c/2}$ where $2^c = 256$, i.e., 16.

This could mean 112/144 rather than 144/112, but we get to pick the monkey's best season.

Please resist the temptation to think to yourselves “This is totally trivial and stupid!”

Please resist the temptation to think to yourselves "This is totally trivial and stupid!"

In reality, the football-monkey attack, far from being trivial and stupid, is how Alfred and I showed that the main theorem of Bellare's Crypto 2006 paper on HMAC is completely worthless in practice.

Please resist the temptation to think to yourselves "This is totally trivial and stupid!"

In reality, the football-monkey attack, far from being trivial and stupid, is how Alfred and I showed that the main theorem of Bellare's Crypto 2006 paper on HMAC is completely worthless in practice.

Namely, the football monkey can be used to show the existence of a generic prf-adversary whose advantage is much, much greater than exhaustive key search.

For any subset S of {1,2,…,c}, any message M in $\{0,1\}^b$ and any bit t, define the bit-valued function $u_{S,M,t}(K)$ for variable K in $\{0,1\}^c$ to be

1 if the XOR sum of the S-indexed bits of f(K,M) is equal to t; 0, otherwise.

For any subset S of {1,2,…,c}, any message M in $\{0,1\}^b$ and any bit t, define the bit-valued function $u_{S,M,t}(K)$ for variable K in $\{0,1\}^c$ to be
1 if the XOR sum of the S-indexed bits of f(K,M) is equal to t; 0, otherwise.

Let (S*,M*,t*) be a fixed triple such that the probability as K varies that $u_{S^*,M^*,t^*}(K)=1$ is maximal.

For any subset S of {1,2,…,c}, any message M in $\{0,1\}^b$ and any bit t, define the bit-valued function $u_{S,M,t}(K)$ for variable K in $\{0,1\}^c$ to be
1 if the XOR sum of the S-indexed bits of f(K,M) is equal to t; 0, otherwise.

Let (S*,M*,t*) be a fixed triple such that the probability as K varies that $u_{S^*,M^*,t^*}(K)=1$ is maximal.

Here's our adversary A: It has (S*,M*,t*) wired into it.

For any subset S of {1,2,…,c}, any message M in $\{0,1\}^b$ and any bit t, define the bit-valued function $u_{S,M,t}(K)$ for variable K in $\{0,1\}^c$ to be
1 if the XOR sum of the S-indexed bits of f(K,M) is equal to t; 0, otherwise.

Let (S*,M*,t*) be a fixed triple such that the probability as K varies that $u_{S^*,M^*,t^*}(K)=1$ is maximal.

Here's our adversary A: It has (S*,M*,t*) wired into it. It makes 1 query, namely M*, to O, which answers O(M*).

For any subset S of {1,2,…,c}, any message M in $\{0,1\}^b$ and any bit t, define the bit-valued function $u_{S,M,t}(K)$ for variable K in $\{0,1\}^c$ to be
1 if the XOR sum of the S-indexed bits of f(K,M) is equal to t; 0, otherwise.

Let (S*,M*,t*) be a fixed triple such that the probability as K varies that $u_{S^*,M^*,t^*}(K)=1$ is maximal.

Here's our adversary A: It has (S*,M*,t*) wired into it. It makes 1 query, namely M*, to O, which answers O(M*). If the XOR sum of the S*-indexed bits of O(M*) is equal to t*, then A guesses that O is really f(K,.); otherwise A guesses that O is random.

Using the standard deviation of a random walk, it's easy to see that one expects A to have advantage at least $2^{-c/2}$.

Using the standard deviation of a random walk, it's easy to see that one expects A to have advantage at least $2^{-c/2}$. Since A's advantage was maximized over 2^{b+c+1} possible triples, this estimate of the expected advantage can be slightly improved using the whole Gaussian distribution, to $(b+c)^{1/2}\ 2^{-c/2}$.

Using the standard deviation of a random walk, it's easy to see that one expects A to have advantage at least $2^{-c/2}$. Since A's advantage was maximized over 2^{b+c+1} possible triples, this estimate of the expected advantage can be slightly improved using the whole Gaussian distribution, to $(b+c)^{1/2}\ 2^{-c/2}$.

To resist the adversary A a compression function f(K,M) would need to have the XOR sum of any subset of its bits much more evenly split between 0's and 1's (for arbitrary fixed M and varying K) than would be expected of a random function.

Using the standard deviation of a random walk, it's easy to see that one expects A to have advantage at least $2^{-c/2}$. Since A's advantage was maximized over 2^{b+c+1} possible triples, this estimate of the expected advantage can be slightly improved using the whole Gaussian distribution, to $(b+c)^{1/2}\ 2^{-c/2}$.

To resist the adversary A a compression function f(K,M) would need to have the XOR sum of any subset of its bits much more evenly split between 0's and 1's (for arbitrary fixed M and varying K) than would be expected of a random function. No one would think that the compression functions used in MD5, SH1, AES256, etc. have such a property. Thus, we can regard A as a generic adversary.

The consequences for the practical interpretation of Bellare's main theorem are catastrophic.

The consequences for the practical interpretation of Bellare's main theorem are catastrophic.

Instead of justifying HMAC security up to 2^{44} queries for MD5 and up to 2^{60} queries for SHA1, because of the football-monkey adversary Bellare's theorem loses content for MD5 after 2^{20} queries and loses content for SHA1 after 2^{28} queries!

From Bellare's 24 February 2012 email to me:

"…if you want your series [of Another Look papers] to gain respect among theoretical cryptographers, it would benefit from reflecting our feedback and being better informed about the basics of the field.

From Bellare's 24 February 2012 email to me:

"…if you want your series [of Another Look papers] to gain respect among theoretical cryptographers, it would benefit from reflecting our feedback and being better informed about the basics of the field. I appreciate this is hard when you are outside the oral culture…."

From Bellare's 24 February 2012 email to me:

"…if you want your series [of Another Look papers] to gain respect among theoretical cryptographers, it would benefit from reflecting our feedback and being better informed about the basics of the field. I appreciate this is hard when you are outside the oral culture….

"Uniform and non-uniform complexity are typically taught in a graduate course in computational complexity theory.

From Bellare's 24 February 2012 email to me:

"…if you want your series [of Another Look papers] to gain respect among theoretical cryptographers, it would benefit from reflecting our feedback and being better informed about the basics of the field. I appreciate this is hard when you are outside the oral culture….

"Uniform and non-uniform complexity are typically taught in a graduate course in computational complexity theory. (They were in the one I took at MIT.)"

Bellare is correct that Alfred and I are “outside the oral culture” of theoretical cryptographers.

Bellare is correct that Alfred and I are "outside the oral culture" of theoretical cryptographers.

And he is correct in supposing that neither of us ever took a complexity theory course at MIT.

Bellare is correct that Alfred and I are "outside the oral culture" of theoretical cryptographers.

And he is correct in supposing that neither of us ever took a complexity theory course at MIT.

But we weren't the ones who failed to understand the dramatic consequences of eschewing Turing machines and opting instead for "Turing machines with advice."

Clint Eastwood

In “Space Cowboys” (Warner Brothers 2000), four old geezers (retired astronauts/engineers) are enlisted to help some young astronauts dismantle a missile-equipped satellite.

In “Space Cowboys” (Warner Brothers 2000), four old geezers (retired astronauts/engineers) are enlisted to help some young astronauts dismantle a missile-equipped satellite.

Some of the drama comes from the tension between the generations.

In “Space Cowboys” (Warner Brothers 2000), four old geezers (retired astronauts/engineers) are enlisted to help some young astronauts dismantle a missile-equipped satellite.

Some of the drama comes from the tension between the generations.

The Clint Eastwood character is called

Dr. Frank Corvin,

and Ethan is one of the young guys.

James Garner, Tommy Lee Jones, Donald Sutherland, Clint Eastwood

Ethan: You're not being very forthcoming on the workings of the guidance systems.

Ethan: You're not being very forthcoming on the workings of the guidance systems.

Frank: Look kid, I just… I've done everything short of calculus instructions to bring you up to speed on this.

Ethan: You're not being very forthcoming on the workings of the guidance systems.

Frank: Look kid, I just… I've done everything short of calculus instructions to bring you up to speed on this. What do you want me to do, draw you a picture? Connect all the little dots?

Ethan: You're not being very forthcoming on the workings of the guidance systems.

Frank: Look kid, I just… I've done everything short of calculus instructions to bring you up to speed on this. What do you want me to do, draw you a picture? Connect all the little dots?

Ethan: Excuse me, I hold two masters degrees from MIT, Dr. Corvin.

Ethan: You're not being very forthcoming on the workings of the guidance systems.

Frank: Look kid, I just… I've done everything short of calculus instructions to bring you up to speed on this. What do you want me to do, draw you a picture? Connect all the little dots?

Ethan: Excuse me, I hold two masters degrees from MIT, Dr. Corvin.

Frank: Maybe you ought to get your money back.

1 ½ –minute Short Course on the Foundations of Computer Science

1 ½ –minute Short Course on the Foundations of Computer Science

An algorithm for solving a problem in the classical sense is a fixed set of instructions that is used for any permissible input of any size.

1 ½ –minute Short Course on the Foundations of Computer Science

An algorithm for solving a problem in the classical sense is a fixed set of instructions that is used for any permissible input of any size.

In contrast, a “non-uniform” algorithm can have a different set of instructions for each input length

1 ½ –minute Short Course on the Foundations of Computer Science

An algorithm for solving a problem in the classical sense is a fixed set of instructions that is used for any permissible input of any size.

In contrast, a "non-uniform" algorithm can have a different set of instructions for each input length; equivalently, it can have an "advice-string" for each input length.

1 ½ –minute Short Course on the Foundations of Computer Science

An algorithm for solving a problem in the classical sense is a fixed set of instructions that is used for any permissible input of any size.

In contrast, a "non-uniform" algorithm can have a different set of instructions for each input length; equivalently, it can have an "advice-string" for each input length.

Note that "input" includes the specification of all parameters needed to describe the problem instance, even if some of them are fixed or vary in a small range in the application we have in mind.

Sometimes polynomial (or other) bounds are imposed on the length of the advice-string.

Sometimes polynomial (or other) bounds are imposed on the length of the advice-string. The complexity class P/poly consists of all problems that can be solved in ptime using psize advice-strings.

Sometimes polynomial (or other) bounds are imposed on the length of the advice-string. The complexity class P/poly consists of all problems that can be solved in ptime using psize advice-strings.

If the term "non-uniform" is used carefully, often there is little difference between asking for a uniform or non-uniform algorithm.

Sometimes polynomial (or other) bounds are imposed on the length of the advice-string. The complexity class P/poly consists of all problems that can be solved in ptime using psize advice-strings.

If the term "non-uniform" is used carefully, often there is little difference between asking for a uniform or non-uniform algorithm.

In his textbook on the foundations of cryptography, Goldreich emphasizes this.

Sometimes polynomial (or other) bounds are imposed on the length of the advice-string. The complexity class P/poly consists of all problems that can be solved in ptime using psize advice-strings.

If the term "non-uniform" is used carefully, often there is little difference between asking for a uniform or non-uniform algorithm.

In his textbook on the foundations of cryptography, Goldreich emphasizes this. He points out that typically a problem has exponentially many inputs of a given bitlength, and a single advice-string is not likely to be helpful for solving exponentially many problem instances.

Take the integer factorization problem.

Take the integer factorization problem. There is no known non-uniform algorithm with polynomial-size advice strings that is faster than all the known uniform algorithms.

Take the integer factorization problem. There is no known non-uniform algorithm with polynomial-size advice strings that is faster than all the known uniform algorithms.

On the other hand, Bernstein and Lange showed that the last statement is no longer true if "polynomial" is replaced by "subexponential".

Take the integer factorization problem. There is no known non-uniform algorithm with polynomial-size advice strings that is faster than all the known uniform algorithms.

On the other hand, Bernstein and Lange showed that the last statement is no longer true if "polynomial" is replaced by "subexponential".

And the picture changes if one is interested in a small subproblem of the original problem that has only a few inputs for each input length.

Take the integer factorization problem. There is no known non-uniform algorithm with polynomial-size advice strings that is faster than all the known uniform algorithms.

On the other hand, Bernstein and Lange showed that the last statement is no longer true if "polynomial" is replaced by "subexponential".

And the picture changes if one is interested in a small subproblem of the original problem that has only a few inputs for each input length.

Example (the Cunningham Project): Factor $b^n \pm 1$ for b=2,3,5,6,7,10,11,12 and large n.

Theorem. **Cunningham Project is in P/poly.**

Theorem. **Cunningham Project is in P/poly.**

Proof. **This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.**

Theorem. **Cunningham Project is in P/poly.**

Proof. **This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.**

Bellare's paper, which nowhere mentions non-uniform complexity, is unclear about exactly what the input is to the prf-security problems in his main theorem.

Theorem. **Cunningham Project is in P/poly.**

Proof. **This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.**

Bellare's paper, which nowhere mentions non-uniform complexity, is unclear about exactly what the input is to the prf-security problems in his main theorem.

The theorem seems to be about a broad class of compression functions.

Theorem. Cunningham Project is in P/poly.

Proof. This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.

Bellare's paper, which nowhere mentions non-uniform complexity, is unclear about exactly what the input is to the prf-security problems in his main theorem.

The theorem seems to be about a broad class of compression functions. However, his advice-string depends on the particular f(K,M).

Theorem. Cunningham Project is in P/poly.

Proof. This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.

Bellare's paper, which nowhere mentions non-uniform complexity, is unclear about exactly what the input is to the prf-security problems in his main theorem.

The theorem seems to be about a broad class of compression functions. However, his advice-string depends on the particular f(K,M). Hence, in order for his proof to be valid in the non-uniform model, he must be proving his theorem for only one or a small number of compression functions for each input length.

Theorem. **Cunningham Project is in P/poly.**

Proof. **This is dumb and totally trivial, since in fact Cunningham Project belongs to Triv/poly.**

Bellare's paper, which nowhere mentions non-uniform complexity, is unclear about exactly what the input is to the prf-security problems in his main theorem.

The theorem seems to be about a broad class of compression functions. However, his advice-string depends on the particular f(K,M). Hence, in order for his proof to be valid in the non-uniform model, he must be proving his theorem for only one or a small number of compression functions for each input length. It's hard to say what he has in mind.

There are many interesting results and open questions relating to elliptic curves that are, loosely speaking, in the “spirit” of non-uniformity. But, with few exceptions, they are not actually related to the non-uniform complexity model.

There are many interesting results and open questions relating to elliptic curves that are, loosely speaking, in the "spirit" of non-uniformity. But, with few exceptions, they are not actually related to the non-uniform complexity model.

1. Oracle complexity of factoring

There are many interesting results and open questions relating to elliptic curves that are, loosely speaking, in the "spirit" of non-uniformity. But, with few exceptions, they are not actually related to the non-uniform complexity model.

1. Oracle complexity of factoring

Ueli Maurer (1995) considered "the problem of factoring integers in polynomial time with the help of an infinitely powerful oracle who answers arbitrary questions with yes or no… The goal is to minimize the number of oracle questions."

Previous non-trivial result (of Rivest-Shamir) for factoring an n-bit integer N: n/3 questions.

Previous non-trivial result (of Rivest-Shamir) for factoring an n-bit integer N: n/3 questions.

Maurer's result: εn questions.

Previous non-trivial result (of Rivest-Shamir) for factoring an n-bit integer N: n/3 questions.

Maurer's result: εn questions.

General idea: Randomly choose a mod-N elliptic curve and ask the oracle how to change the bits in the coefficients of the curve's equation so as to get a curve that has smooth order mod p (where p is a prime factor of N). After that, Elliptic Curve Factorization will take only polynomial time.

Previous non-trivial result (of Rivest-Shamir) for factoring an n-bit integer N: n/3 questions.

Maurer's result: εn questions.

General idea: Randomly choose a mod-N elliptic curve and ask the oracle how to change the bits in the coefficients of the curve's equation so as to get a curve that has smooth order mod p (where p is a prime factor of N). After that, Elliptic Curve Factorization will take only polynomial time.

This is a nice result even though it has no use for practical cryptography.

2. Maurer-Wolf's DHP-DLP equivalence

See: *SIAM J. Computing* 28 (1999), pp. 1689-1721.

2. Maurer-Wolf's DHP-DLP equivalence

See: *SIAM J. Computing* 28 (1999), pp. 1689-1721.

Again using elliptic curves of smooth order, they show that "for every cyclic group G there exists a small piece of information, which depends only on the order of G, that makes breaking the Diffie-Hellman protocol and computing discrete logarithms equivalent in G."

2. Maurer-Wolf's DHP-DLP equivalence

See: *SIAM J. Computing* 28 (1999), pp. 1689-1721.

Again using elliptic curves of smooth order, they show that "for every cyclic group G there exists a small piece of information, which depends only on the order of G, that makes breaking the Diffie-Hellman protocol and computing discrete logarithms equivalent in G."

This is a wonderful result, but they incorrectly term it "non-uniform equivalence". The advice-string depends on the order of G, not just on the bitlength of the order of G.

As the above examples show, it might be very interesting to consider algorithms with advice-strings even if the algorithms are not non-uniform.

As the above examples show, it might be very interesting to consider algorithms with advice-strings even if the algorithms are not non-uniform.

Warning about weird terminology: "non-uniform" does not mean "not uniform." An algorithm can certainly fail to be either uniform or non-uniform.

As the above examples show, it might be very interesting to consider algorithms with advice-strings even if the algorithms are not non-uniform.

Warning about weird terminology: "non-uniform" does not mean "not uniform." An algorithm can certainly fail to be either uniform or non-uniform.

Even worse, every uniform algorithm is also a non-uniform algorithm (with empty advice-strings).

<u>Open problems</u>. (1) Give a ptime algorithm that, given any finite field F_q and a psize advice-string that depends only on q, computes discrete logs in F_q.

<u>Open problems</u>. (1) Give a ptime algorithm that, given any finite field F_q and a psize advice-string that depends only on q, computes discrete logs in F_q.

(2) Give a polynomial-time algorithm that, given any elliptic curve E defined over a finite field and a polynomial-size advice-string that depends only on E, computes discrete logs on E.

Open problems. **(1) Give a ptime algorithm that, given any finite field F_q and a psize advice-string that depends only on q, computes discrete logs in F_q.**

(2) Give a polynomial-time algorithm that, given any elliptic curve E defined over a finite field and a polynomial-size advice-string that depends only on E, computes discrete logs on E.

(3) The same as (2) with either or both occurrences of "polynomial" replaced by "subexponential", or by $N^{1/4}$, where N is the order of the group E.

Note that Bernstein-Lange have a result of the form (3).

Note that Bernstein-Lange have a result of the form (3).

Results of this type do not give non-uniform algorithms for the ECDLP.

Note that Bernstein-Lange have a result of the form (3).

Results of this type do not give non-uniform algorithms for the ECDLP.

But suppose we are interested only in a restricted problem, such as $\mathrm{ECDLP}_{\mathrm{ABC}}$, the problem of finding discrete logs on anomalous binary curves.

Note that Bernstein-Lange have a result of the form (3).

Results of this type do not give non-uniform algorithms for the ECDLP.

But suppose we are interested only in a restricted problem, such as $ECDLP_{ABC}$, the problem of finding discrete logs on anomalous binary curves. There is only one ABC curve for each even-degree extension of F_2 and two for each odd-degree extension of F_2. Then solutions to (2) or (3) will give non-uniform algorithms.

Finally, I want to mention a valiant but unsuccessful attempt to solve an open problem similar to (1) above but for the decision Diffie-Hellman problem rather than the discrete log problem.

3. The torsion-subgroup “non-uniform” attack on Decision D-H

Q. Cheng and S. Uchiyama, *Nonuniform polynomial time algorithm to solve decisional Diffie-Hellman in finite fields under conjecture,* CT-RSA 2002.

3. The torsion-subgroup "non-uniform" attack on Decision D-H

Q. Cheng and S. Uchiyama, *Nonuniform polynomial time algorithm to solve decisional Diffie-Hellman in finite fields under conjecture,* CT-RSA 2002.

They attempted to solve the open question:
Is it possible, given a polynomial-size hint depending on the primes p and n (where n|(p-1)) but not on the rest of the input, to solve any instance of DDH in the order-n subgroup of $(F_p)^*$?

Note that even if they could give an affirmative answer to that question – which they claimed to have done under a plausible conjecture – it would not have given a non-uniform ptime DDH algorithm.

Note that even if they could give an affirmative answer to that question – which they claimed to have done under a plausible conjecture – it would not have given a non-uniform ptime DDH algorithm.

The "hint" or advice-string for a non-uniform algorithm must be the same for all p and n of given bitlengths.

Note that even if they could give an affirmative answer to that question – which they claimed to have done under a plausible conjecture – it would not have given a non-uniform ptime DDH algorithm.

The "hint" or advice-string for a non-uniform algorithm must be the same for all p and n of given bitlengths.

Alfred and I analyzed their algorithm in *Math. Comp.* 73 (2004), pp. 2027-2041.

The advice-string they needed was an elliptic curve defined over a number field K that has a torsion-point of order n. The only evidence for their conjecture that such curves exist – with psize [K:Q], psize generating polynomial for K, psize coefficients of the curve, etc. – is the fact that mathematicians have not yet been able to prove that they do not exist.

The advice-string they needed was an elliptic curve defined over a number field K that has a torsion-point of order n. The only evidence for their conjecture that such curves exist – with psize [K:Q], psize generating polynomial for K, psize coefficients of the curve, etc. – is the fact that mathematicians have not yet been able to prove that they do not exist.

However, there are heuristic arguments and indirect evidence (partial results) that make it seem extremely doubtful that such curves exist.

From my article “Good and bad uses of elliptic curves in cryptography” in the Moscow Math Journal (2002 Manin-birthday issue):

From my article "Good and bad uses of elliptic curves in cryptography" in the Moscow Math Journal (2002 Manin-birthday issue):

"Of course, in the absence of a proof that such a point [modular point corresponding to the required high-torsion curve] cannot exist, one is free to conjecture whatever one's heart desires, just as one is free to conjecture the existence of little green men walking across the ice fields of Ganymede."

From my article "Good and bad uses of elliptic curves in cryptography" in the Moscow Math Journal (2002 Manin-birthday issue):

"Of course, in the absence of a proof that such a point [modular point corresponding to the required high-torsion curve] cannot exist, one is free to conjecture whatever one's heart desires, just as one is free to conjecture the existence of little green men walking across the ice fields of Ganymede."

Ganymede

Conclusion

From a practice-oriented standpoint, non-uniform complexity (as well as other notions of algorithms with advice-strings) are not of much use.

Conclusion

From a practice-oriented standpoint, non-uniform complexity (as well as other notions of algorithms with advice-strings) are not of much use.

The advice-strings do a poor job of modeling anything practical – for example,

- **precomputation,**
- **side-channel leakage.**

Conclusion

From a practice-oriented standpoint, non-uniform complexity (as well as other notions of algorithms with advice-strings) are not of much use.

The advice-strings do a poor job of modeling anything practical – for example,

- **precomputation,**
- **side-channel leakage.**

What they do model is magic –

Conclusion

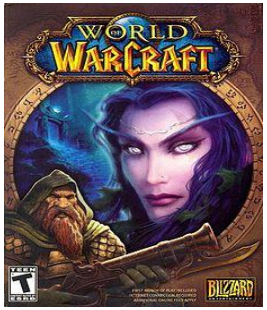
From a practice-oriented standpoint, non-uniform complexity (as well as other notions of algorithms with advice-strings) are not of much use.

The advice-strings do a poor job of modeling anything practical – for example,

- **precomputation,**
- **side-channel leakage.**

What they do model is magic –

- **World of Warcraft,**



Conclusion

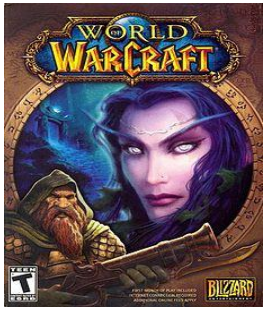
From a practice-oriented standpoint, non-uniform complexity (as well as other notions of algorithms with advice-strings) are not of much use.

The advice-strings do a poor job of modeling anything practical – for example,

- **precomputation,**
- **side-channel leakage.**

What they do model is magic –

- **World of Warcraft,**
- **Harry Potter.**



Nevertheless, results and open questions involving algorithms with advice-strings are sometimes cute and stimulating to think about – for instance, Maurer's work on the oracle complexity of factoring.

Nevertheless, results and open questions involving algorithms with advice-strings are sometimes cute and stimulating to think about – for instance, Maurer's work on the oracle complexity of factoring.

There is nothing shameful about working on problems that have great aesthetic appeal and zero practical utility.

Nevertheless, results and open questions involving algorithms with advice-strings are sometimes cute and stimulating to think about – for instance, Maurer's work on the oracle complexity of factoring.

There is nothing shameful about working on problems that have great aesthetic appeal and zero practical utility.

Following G. H. Hardy, we should be happy to be working on something "gentle and clean".

Nevertheless, results and open questions involving algorithms with advice-strings are sometimes cute and stimulating to think about – for instance, Maurer's work on the oracle complexity of factoring.

There is nothing shameful about working on problems that have great aesthetic appeal and zero practical utility.

Following G. H. Hardy, we should be happy to be working on something "gentle and clean".

After all, Fermat's Last Theorem has absolutely no practical utility whatsoever.

There is nothing at all wrong with a result that is proved in the non-uniform model of complexity, or makes assumptions that are in the "spirit" of non-uniformity —

There is nothing at all wrong with a result that is proved in the non-uniform model of complexity, or makes assumptions that are in the "spirit" of non-uniformity —

provided that no claim is made about its practicality.