RESEARCH ANNOUNCEMENT: STRONGER SECURITY BOUNDS FOR WEGMAN-CARTER-SHOUP AUTHENTICATORS

DANIEL J. BERNSTEIN

Abstract. Shoup proved that various message-authentication codes of the form $(n, m) \mapsto h(m) + f(n)$ are secure against all attacks that see at most $\sqrt{1/\epsilon}$ authenticated messages. Here m is a message, n is a nonce chosen from a public group G, f is a secret uniform random permutation of G, h is a secret random function, and ϵ is a differential probability associated with h.

Shoup's result implies that if AES is secure then various state-of-the-art message-authentication codes of the form $(n, m) \mapsto h(m) + \mathrm{AES}_k(n)$ are secure up to $\sqrt{1/\epsilon}$ authenticated messages. Unfortunately, $\sqrt{1/\epsilon}$ is only about 2^{50} for some state-of-the-art systems, so Shoup's result provides no guarantees for long-term keys.

This paper proves that security of the same systems is retained up to $\sqrt{\#G}$ authenticated messages. In a typical state-of-the-art system, $\sqrt{\#G}$ is 2^{64}. The heart of the paper is a very general "one-sided" security theorem: $(n, m) \mapsto h(m) + f(n)$ is secure if there are small upper bounds on differential probabilities for h and on interpolation probabilities for f.

1. Introduction

This paper proves that various state-of-the-art 128-bit authenticators are secure against all attacks that see at most 2^{64} authenticated messages. Previous proofs broke down at a smaller number of messages, often below 2^{50}.

A typical example. Consider the well-known polynomial-evaluation message-authentication code over a field of size 2^{128}.

Each message is a polynomial over the field with constant coefficient 0. The sender's nth message, say m_n, is transmitted as $(n, m_n, m_n(r) + f(n))$; here r and f are secrets shared by the sender and the receiver. What is the attacker's chance of successfully forging a message?

It is easy to prove information-theoretic security of this system if r and f are independent, r is a uniform random element of the field, and f is a uniform random function from $\{n\}$ to the field—in other words, if $r, f(1), f(2), \ldots$ are independent uniform random elements of the field. The attacker's chance of success is at most

Date: 2004.10.19. Permanent ID of this document: `2d603727f69542f30f7da2832240c1ad`. This is a preliminary version meant to announce ideas; it will be replaced by a final version meant to record the ideas for posterity. There may be big changes before the final version. Future readers should not be forced to look at preliminary versions, unless they want to check historical credits; if you cite a preliminary version, please repeat all ideas that you are using from it, so that the reader can skip it.

2000 *Mathematics Subject Classification.* Primary 94A60; Secondary 94A62, 94A15.

The author was supported by the National Science Foundation under grant CCR–9983950, and by the Alfred P. Sloan Foundation.

$LD/2^{128}$, where L is the maximum degree of a message and D is the number of forgeries attempted. The point is that $m_n(r) + f(n)$ leaks no information about $m_n(r)$.

What if f is a uniform random *injective* function—in other words, $f(1), f(2), \ldots$ are chosen to be distinct? If the sender transmits only C messages, where C is small, then $f(1), f(2), \ldots, f(C)$ are *nearly* independent, and one can easily prove that the attacker's chance of success is at most $LD/2^{128} + C(C-1)/2^{129}$; but this bound becomes useless as C approaches 2^{64}. Shoup proved a better bound in [6, Theorem 2]: the attacker's chance of success is at most $2LD/2^{128}$ if $C \le 2^{64}/\sqrt{L}$. This paper eliminates the $\sqrt{L}$ denominator: the attacker's chance of success is below $1.002LD/2^{128}$ if $C \le 2^{60}$, and below $1.7LD/2^{128}$ if $C \le 2^{64}$, and below $3000LD/2^{128}$ if $C \le 2^{66}$.

For example, say the sender authenticates $C = 2^{60}$ messages, the attacker tries $D = 2^{60}$ forgeries, and the maximum message degree is $L = 2^{16}$. The easy bound is about $1/2^9$, which is not at all comforting. Shoup's bound is inapplicable. The bound in this paper is $1.002/2^{51}$.

Consequences for AES-based authenticators. Despite the high speed and information-theoretic security of $m_n(r)+f(n)$, users often prefer $m_n(r)+\text{AES}_k(n)$. The point is that r, k occupy only 32 bytes, whereas $r, f(1), f(2), \ldots$ occupy an additional 16 bytes for each message.

The attacker's success chance against $m_n(r) + \text{AES}_k(n)$ is bounded by the sum of two terms: first, the attacker's success chance against $m_n(r) + f(n)$; second, the attacker's chance of distinguishing AES_k from f. In particular:

- Take f to be a uniform random function. In this case, the first term—the attacker's success chance against $m_n(r) + f(n)$—is easily proven to be small. Unfortunately, the second term becomes large as C approaches 2^{64}: the attacker can distinguish AES_k from f with probability $C(C-1)/2^{129}$ by looking for collisions.
- Take f to be a uniform random injective function. In this case, the first term is small, even for $C = 2^{64}$; that is the point of this paper. The second term is conjectured to also be small: it appears to be extremely difficult to distinguish AES_k from f, even after 2^{65} chosen inputs. "Indistinguishability from a uniform random permutation" was an explicit design goal for AES.

In short, this paper guarantees that $m_n(r) + \text{AES}_k(n)$ is as secure as AES up to 2^{64} messages. The best previous results did not handle nearly as many messages.

The importance in this context of uniform random injective functions, as opposed to uniform random functions, was pointed out by Shoup in [6, Section 1].

Generalization. This paper considers much more general message-authentication codes of the form $(n, m) \mapsto h(m)+f(n)$. The main theorem of this paper, Theorem 5.1, is that $h(m) + f(n)$ is secure if (1) differential probabilities for h are small and (2) interpolation probabilities for f are small.

In particular, assume that f is a uniform random injective function from the set of nonces to a finite commutative group G, and that the differential probabilities for h are small. Then $h(m) + f(n)$ is secure against all attacks that see at most $\sqrt{\#G}$ authenticated messages. Consequently $h(m) + \text{AES}_k(n)$ is secure against any attacker who cannot break AES and who sees at most $\sqrt{\#G}$ authenticated messages.

The form $h(m) \oplus f(n)$ for an authenticator, where f is a uniform random function, was introduced by Wegman and Carter in [8, Section 4]. Here $\oplus$ is vector addition modulo 2. Brassard in [2] considered $h(m) \oplus f(n)$ where f is a random injective function determined by a short key, such as AES_k. Shoup in [6], as discussed above, considered $h(m) \oplus f(n)$ where f is a uniform random injective function. The more general shape $h(m) + f(n)$, where $+$ can be any commutative group operation, is helpful for accommodating functions that rely on addition in large characteristic rather than characteristic 2—in particular, functions that rely on the high-speed multiplication circuits included in common processors.

All of the proofs in the literature rely on two-sided bounds for the interpolation probabilities for f. One computes lower bounds on the probability of any particular sequence of authenticators; one computes nearby upper bounds on the probability of that sequence of authenticators given h; one deduces that the authenticators reveal very little information about h, and hence very little information about the authenticator for a new message. See, e.g., [8, Section 4, Theorem] and [6, Appendix A, Lemma 1]. The heart of the improvement in this paper is a new "one-sided" proof strategy that moves directly from upper bounds for f and h to upper bounds on the attacker's chance of success.

2. Protocol

This section describes a very general message-authentication protocol. Section 3 formalizes the notion of an attack on the protocol. Section 5 analyzes the success chance of all attacks.

The protocol has several parameters:

- G, a finite commutative group of **authenticators**. I will always write the group operation as $+$. (More general groups, or even loops, would suffice, but I see no application of the extra generality.) Typical example: G is the set of 16-byte strings, with the group operation being exclusive-or. Another example: G is the set $\{0, 1, 2, \ldots, 2^{128} - 1\}$, with the group operation being addition modulo 2^{128}.
- M, a nonempty set of **messages**. Typical example: M is the set of all strings of bytes. Another example: M is the set of all strings of at most 1024 bytes.
- N, a finite set of **nonces**, with $\#N \le \#G$. Typical example: N is the set $\{1, 2, 3, \ldots, 2^{32} - 1\}$. Another example: N is the set of 16-byte strings.

The protocol has several participants:

- A **message generator** creates messages.
- A **nonce generator** accepts messages from the message generator and attaches a nonce n to each message m. The nonce generator must never use the same nonce for two different messages: if it generates (n_1, m_1) and (n_2, m_2), and if $m_1 \ne m_2$, then n_1 must not equal n_2. This rule is automatically satisfied if the nonce generator uses nonce 1 for the first message, nonce 2 for the second message, etc.
- A **sender** accepts pairs (n, m) from the nonce generator and attaches an authenticator a to each pair, as discussed below.
- A **network** accepts a sequence of vectors (n, m, a) from the sender and transmits a sequence of vectors (n', m', a'). Perhaps the sequence of vectors transmitted is the same as the sequence of vectors sent; perhaps not.

- A **receiver** receives vectors (n', m', a') from the network. It accepts (n', m') if a' is the authenticator that the sender would have attached to (n', m'); otherwise it discards (n', m').

If the network transmits exactly what the sender sent, then the pairs (n, m) accepted by the receiver are exactly the pairs (n, m) given to the sender; but what if the network makes changes? The objective of the protocol is **forgery elimination**: ensuring that each pair (n', m') accepted by the receiver is one of the pairs (n, m) that was authenticated by the sender.

One could ask for additional protocol features:

- The receiver should notice if the network repeats messages or transmits messages out of order. One way to do this is for the nonce generator to use increasing nonces (in some specified ordering of the set N), and for the receiver to discard (n', m', a') unless n' is larger than the last accepted nonce.
- The receiver should notice if the network loses a message. There's no way to recover if the network is losing all messages, but there are retransmission protocols that eventually succeed in transmitting all data if the network delivers (e.g.) 1% of all messages.

But this paper focuses on the cryptographic problem of forgery elimination.

The sender's authenticator for a pair (n, m) is $h(m) + f(n)$: i.e., the sender gives $(n, m, h(m, f(n)))$ to the network. Here h is a random function from M to G, and f is a random function from N to G. The pair (f, h) is a secret shared by the sender and receiver; this means that the actions of the message generator, nonce generator, and network are independent of (f, h). In particular, if the message generator encrypts messages, it does so using a key independent of (f, h). The proof strategy in this paper can be extended to cover protocols that reuse f for encryption, as long as separate f inputs are used for encryption and for authentication; but that extension is not included in the statement of Theorem 5.1.

Warning: "Random" and "uniform random" and "independent uniform random" do not mean the same thing. For example, if k is a uniform random 16-byte string, then $(k, 0)$ is a non-uniform random 17-byte string; AES_k is a non-uniform random permutation of the set of 16-byte strings; $k[0]$, the first byte of k, is a uniform random byte; $k[0]$, $k[1]$, and $k[2]$ are independent uniform random bytes; $k[0]$, $k[1]$, and $k[0] \oplus k[1]$ are non-independent uniform random bytes; $(k[0], 0)$ and $(k[1], 0)$ are independent non-uniform random 2-byte strings. I realize that the word "random" is sometimes used to mean "uniform random, independent of everything else," but a more careful use of terminology is helpful in stating and proving theorems.

3. Attacks

The combined behavior of the message generator, nonce generator, and network is called an "attack." The attack creates messages; it creates nonces, subject to the rule that nonces never repeat; it inspects the authenticators provided by the sender; and it provides some number of forgeries to the receiver.

More formally: An **attack** is an algorithm given oracle access to a function S. The algorithm feeds a nonce n_1 and message m_1 to the oracle. It receives an authenticator $a_1 = S(n_1, m_1)$. It then feeds a nonce n_2 and message m_2 to the oracle, obeying the rule that $n_2 \neq n_1$ if $m_2 \neq m_1$. It receives an authenticator $a_2 = S(n_2, m_2)$. It then feeds a nonce n_3 and message m_3 to the oracle, obeying the rule that $n_3 \neq n_1$ if $m_3 \neq m_1$, and the rule that $n_3 \neq n_2$ if $m_3 \neq m_2$. It receives an authenticator $a_3 = S(n_3, m_3)$. It continues for any number of messages. It then prints some number of **forgery attempts** (n', m', a').

The attack **succeeds against** S if at least one forgery attempt (m', n', a') has $a' = S(n', m')$ with $(n', m') \notin \{(n_1, m_1), (n_2, m_2), (n_3, m_3), \ldots\}$.

Is there an attack that succeeds against $(n, m) \mapsto h(m) + f(n)$ with noticeable probability? Theorem 5.1 states, under certain assumptions on f and h, that the answer is no. The receiver is overwhelmingly likely to discard every forgery—no matter how the message generator chooses messages, now matter how the nonce generator chooses unique nonces, and no matter how the network chooses forgeries. (However, if the nonce generator repeats a nonce, all bets are off!)

The rest of this section discusses the strength of this theorem, under the same assumptions on f and h.

Forgeries versus selective forgeries. A selective forgery is a forged message chosen in advance by the attacker. Some protocols prevent selective forgeries but allow attackers to compute authenticators for random-looking messages. These protocols assume—often incorrectly—that random-looking messages will not cause any damage. In contrast, $h(m) + f(n)$ rejects *all* forgeries.

Attacks versus blind attacks. Some protocols prevent blind attacks but allow forgeries when attackers can inspect authenticated messages. (Trivial example: use a secret password as an authenticator for every message.) In contrast, $h(m) + f(n)$ rejects all forgeries even after the attacker sees a large number of authenticated messages. This paper does not rely on secrecy.

Chosen messages versus known messages. Some protocols are secure for some message generators but are insecure for others. An attacker who can influence the message generator can often obtain enough information to forge messages. In contrast, $h(m) + f(n)$ rejects all forgeries no matter what the message generator does.

Of course, if an attacker can somehow convince the message generator to produce a message, then he does not need to forge an authenticator for that message. An easily corrupted message generator is often a problem. It is, however, not the cryptographic problem considered in this paper.

Receiver interaction. In Section 2, the receiver is not a source of information. However, when the same protocol is placed into a larger context, the receiver often becomes a source of information, revealing to the attacker whether a forgery was accepted.

One can expand the notion of “attack” to allow interaction with a verification oracle. However, this expansion makes no difference in the attack's success chance. An attack that interacts with the receiver has the same success chance as an attack that skips the interactions and simply assumes that all the forgeries are rejected. This type of interaction can change the *number* of successful forgeries if the attacker succeeds, but this paper guarantees that the attacker will not succeed in the first place.

4. Interpolation probabilities

Let f be a random function from N to G. The hypothesis on f in Section 5 is that f has maximum k-interpolation probability on the scale of $1/\#G^k$, for various $k \in \{0, 1, \ldots, \#N\}$. Here the **maximum k-interpolation probability of f** is the maximum, for all $x_1, x_2, \ldots, x_k \in G$ and all distinct $n_1, n_2, \ldots, n_k \in N$, of the probability that $(f(n_1), f(n_2), \ldots, f(n_k)) = (x_1, x_2, \ldots, x_k)$.

This section proves that this condition is satisfied by a uniform random function and by a uniform random injective function.

Theorem 4.1. *Let f be a uniform random function from N to G. Then f has maximum k-interpolation probability $1/\#G^k$ for each $k \in \{0, 1, \ldots, \#N\}$.*

Proof. The probability that $(f(n_1), f(n_2), \ldots, f(n_k)) = (x_1, x_2, \ldots, x_k)$ is exactly $1/\#G^k$. □

Theorem 4.2. *Let f be a uniform random injective function from N to G. Then f has maximum k-interpolation probability at most $(1 - (k-1)/\#G)^{-k/2}/\#G^k$ for each $k \in \{0, 1, \ldots, \#N\}$.*

Proof. Fix distinct $n_1, n_2, \ldots, n_k \in N$. Fix $x_1, x_2, \ldots, x_k \in G$.

Case 1: There are collisions in $x_1, x_2, \ldots, x_k$. Then $(f(n_1), f(n_2), \ldots, f(n_k)) = (x_1, x_2, \ldots, x_k)$ with probability 0.

Case 2: There are no collisions. Then $f(n_1) = x_1$ with probability $1/\#G$; if that happens then $f(n_2) = x_2$ with conditional probability $1/(\#G - 1)$; if that happens then $f(n_3) = x_3$ with conditional probability $1/(\#G - 2)$; and so on through $f(n_k) = x_k$. The probability that $(f(n_1), f(n_2), \ldots, f(n_k)) = (x_1, x_2, \ldots, x_k)$ is $\prod_{0 \le i \le k-1} 1/(\#G - i)$, with square $\prod_{0 \le i \le k-1} 1/(\#G - i)(\#G - (k - 1 - i)) \le \prod_{0 \le i \le k-1} 1/(\#G)^2(1 - (k-1)/\#G) = (1 - (k-1)/\#G)^{-k}/(\#G)^{2k}$. □

5. The main theorem

Theorem 5.1. *Let h be a random function from M to G. Let f be a random function from N to G. Let C and D be positive integers. Assume that $C+1 \le \#N$. Assume, for all $g \in G$ and all distinct $m, m' \in M$, that $h(m) = h(m') + g$ with probability at most ϵ. Assume that f has maximum C-interpolation probability at most $\delta/\#G^C$ and maximum $(C+1)$-interpolation probability at most $\delta\epsilon/\#G^C$. Assume that h and f are independent. Then any attack that performs at most C distinct oracle queries and at most D forgery attempts succeeds against $(n, m) \mapsto h(m) + f(n)$ with probability at most $D\delta\epsilon$.*

Proof. It suffices to show that each forgery attempt succeeds with probability at most $\delta\epsilon$. Assume from now on that the attack makes exactly one forgery attempt.

If the attack performs fewer than C distinct oracle queries, modify it to perform additional oracle queries with new nonces and to discard the results; new nonces are available since $\#N \ge C$, and at least one message is available since $\#M \ge 1$. Assume from now on that the attack makes exactly C distinct oracle queries.

If the attack might repeat oracle queries, modify it to cache oracle queries and responses. Assume from now on that the attack does not repeat queries.

Write (n_i, m_i) for the ith oracle query. Then $n_1, n_2, \ldots, n_C$ are distinct. Write a_i for the ith oracle response, when the attack is applied to $(n, m) \mapsto h(m) + f(n)$; then $a_i = h(m_i) + f(n_i)$. Write (n', m', a') for the attempted forgery.

Everything that the attack does is determined by (1) an infinite sequence b of coin flips, by definition independent of h and f, and (2) the sequence of oracle responses $a_1, a_2, \ldots, a_C$. In particular, $n_1, n_2, \ldots, n_C, m_1, m_2, \ldots, m_C, n', m', a'$ are equal to various functions evaluated at $b, a_1, a_2, \ldots, a_C$. Furthermore, $f(n_i)$ is determined by a_i and $h(m_i)$, so $f(n_i)$ is equal to a function evaluated at $h, b, a_1, a_2, \ldots, a_C$.

Fix $(g_1, g_2, \ldots, g_C) \in G^C$. Consider the event that (n', m', a') is a successful forgery and $(a_1, a_2, \ldots, a_C) = (g_1, g_2, \ldots, g_C)$. It suffices to show that this event has probability at most $\delta\epsilon/\#G^C$.

Define p as the probability that b satisfies the following (measurable) constraint: if $(a_1, a_2, \ldots, a_C) = (g_1, g_2, \ldots, g_C)$ then $n' \notin \{n_1, n_2, \ldots, n_C\}$. I claim, for each b satisfying the constraint and for each h, that f has conditional probability at most $\delta\epsilon/\#G^C$ of making the attack work.

Indeed, assume that b satisfies the constraint, that (n', m', a') is a successful forgery, and that $(a_1, a_2, \ldots, a_C) = (g_1, g_2, \ldots, g_C)$. Then $\#\{n_1, n_2, \ldots, n_C, n'\} = C + 1$, and $f(n_1), f(n_2), \ldots, f(n_C), f(n')$ match various functions evaluated at $h, b, g_1, g_2, \ldots, g_C$. By hypothesis, f is independent of h; f is also independent of b; and $g_1, g_2, \ldots, g_C$ are fixed. The conditional probability of f interpolating these values is at most the maximum $(C+1)$-interpolation probability of f, which by hypothesis is at most $\delta\epsilon/\#G^C$.

I also claim, for each b *not* satisfying the constraint, that h has conditional probability at most ϵ of satisfying a necessary differential condition; and, for each b and each qualifying h, that f has conditional probability at most $\delta/\#G^C$ of making the attack work.

Indeed, assume that b does not satisfy the constraint, that $(a_1, a_2, \ldots, a_C) = (g_1, g_2, \ldots, g_C)$, and that (n', m', a') is a successful forgery. Then $n' = n_i$ for a unique i; $a' = h(m') + f(n_i)$; $m' \neq m_i$; and $a_i = h(m_i) + f(n_i)$. Now $h(m_i) - h(m') = a_i - a'$; the quantities m_i, m' and $a_i - a'$ match various functions evaluated at $b, g_1, g_2, \ldots, g_C$, and thus are independent of h; by hypothesis, h satisfies the condition $h(m_i) - h(m') = a_i - a'$ with probability at most ϵ. Furthermore, $f(n_1), f(n_2), \ldots, f(n_C)$ match various functions evaluated at $h, b, g_1, g_2, \ldots, g_C$; f is independent of $h, b, g_1, g_2, \ldots, g_C$; the conditional probability of f interpolating these values is at most the maximum C-interpolation probability of f, which by hypothesis is at most $\delta/\#G^C$.

The total probability of success is at most $(p)(\delta\epsilon/\#G^C) + (1-p)(\epsilon)(\delta/\#G^C) = \delta\epsilon/\#G^C$. □

Theorem 5.2. *Let h be a random function from M to G. Let f be a uniform random function from N to G. Let C and D be positive integers. Assume that $C + 1 \leq \#N$. Assume, for all $g \in G$ and all distinct $m, m' \in M$, that $h(m) = h(m') + g$ with probability at most ϵ. Assume that h and f are independent. Then any attack that performs at most C distinct oracle queries and at most D forgery attempts succeeds against $(n, m) \mapsto h(m) + f(n)$ with probability at most $D \max\{\epsilon, 1/\#G\}$.*

Proof. Take $\delta = \max\{1, 1/\epsilon\#G\}$. Apply Theorem 4.1 and Theorem 5.1. □

Theorem 5.3. *Let h be a random function from M to G. Let f be a uniform random injective function from N to G. Let C and D be positive integers. Assume that $C + 1 \leq \#N$. Assume, for all $g \in G$ and all distinct $m, m' \in M$, that $h(m) = h(m') + g$ with probability at most ϵ. Assume that h and f are independent. Then any attack that performs at most C distinct oracle queries and at most D*

forgery attempts succeeds against $(n, m) \mapsto h(m) + f(n)$ *with probability at most* $D(1 - C/\#G)^{-(C+1)/2} \max\{\epsilon, 1/\#G\}$.

In particular, if $C = \lfloor \sqrt{\#G} \rfloor$, then the extra factor $(1 - C/\#G)^{-(C+1)/2}$ is below 1.7 for all reasonably large G.

Proof. Take $\delta = (1 - C/\#G)^{-(C+1)/2} \max\{1, 1/\epsilon\#G\}$. Apply Theorem 4.2 and Theorem 5.1. □

References

[1] —, *20th annual symposium on foundations of computer science*, IEEE Computer Society, New York, 1979. MR 82a:68004.

[2] Gilles Brassard, *On computationally secure authentication tags requiring short secret shared keys*, in [3] (1983), 79–86. URL: `http://cr.yp.to/bib/entries.html#1983/brassard`.

[3] David Chaum, Ronald L. Rivest, Alan T. Sherman (editors), *Advances in cryptology: proceedings of Crypto 82*, Plenum Press, New York, 1983. ISBN 0–306–41366–3. MR 84j:94004.

[4] Neal Koblitz (editor), *Advances in cryptology—CRYPTO '96*, Lecture Notes in Computer Science, 1109, Springer-Verlag, Berlin, 1996.

[5] Victor Shoup, *On fast and provably secure message authentication based on universal hashing*, in [4] (1996), 313–328; see also newer version [6].

[6] Victor Shoup, *On fast and provably secure message authentication based on universal hashing* (1996); see also older version [5]. URL: `http://www.shoup.net/papers`.

[7] Mark N. Wegman, J. Lawrence Carter, *New classes and applications of hash functions*, in [1] (1979), 175–182; see also newer version [8]. URL: `http://cr.yp.to/bib/entries.html#1979/wegman`.

[8] Mark N. Wegman, J. Lawrence Carter, *New hash functions and their use in authentication and set equality*, Journal of Computer and System Sciences **22** (1981), 265–279; see also older version [7]. ISSN 0022–0000. MR 82i:68017. URL: `http://cr.yp.to/bib/entries.html#1981/wegman`.

Department of Mathematics, Statistics, and Computer Science (M/C 249), The University of Illinois at Chicago, Chicago, IL 60607–7045

E-mail address: `djb@cr.yp.to`