A moving target and shuffle-decimation quarantine strategy to combat persistent DDoS attacks on a distributed, indirection server system in the cloud

George Kesidis, Yuquan Shan, Angelos Stavrou, Dan Fleck, Takis Konstantopoulos
The Pennsylvania State University, George Mason University, Uppsala University
University Park, PA, Fairfax, VA, Sweden

CSE Dept, PSU, Technical Report CSE-16-012
http://www.cse.psu.edu/research/publications/tech-reports/2016/CSE-16-012.pdf
Oct 31, 2016

I. INTRODUCTION

Consider a system of clients, indirection/proxy servers and worker/application servers (replicas) controlled either by a central coordination server (e.g., via Round-robin DNS) or in distributed fashion by the proxies themselves [2], [3]. The clients reach the proxies via the public commodity Internet. The proxies and replicas could be implemented in VMs (or in containers within VMs) of one or more datacenters. The proxies in turn assign clients to replicas to fulfill their requests. The assignment is nominally via a DHT. The clients never know the IP addresses of the replicas, only the proxies.

Threats can target either proxies (e.g., a volumetric DDoS attack, preliminary probing) or replicas (protocol attack or a localized DoS attack such as Slowloris, i.e., a single attacker can bring down the replica). A key assumption is that a server under attack cannot determine which of its assigned client(s) are responsible. This said, it can be reliably determined whether a server is under attack either through the use of probing “canary” clients that can determine when response times of their mock workloads have grown too high, or by the OS of a VM (or hypervisor of a physical server) managing a number of containers (or VMs) in which proxies or replicas operate. New clients are never added to servers (either proxies or replicas) deemed under attack.

Some attacking clients are simple bots that cannot cope with reassignment to a new server. For such bots, continual shuffling of clients to servers is a proactive defense. We focus on other malware that is more resilient and can continue its attack even after it is transferred to another server (or the server just modifies its IP address).

Both proactive moving target and reactive quarantine (shuffling, fission) defenses are explored herein. For the proxy servers, we study the benefits and costs of

- (proactive) moving target - impact of continually changing assigned IP addresses on the situational awareness of the probing (reconnoitering) botnet regarding the set of “fresh” (visible) IP addresses of the targeted multiserver system
- (reactive) shuffle defense under the proxies’ limited volumetric attack tolerance - generalized $r$-associated Stirling numbers of the second kind, post-attack shuffling of client-to-server assignments to quarantine attackers

For the replicas (application servers), we study

- (reactive) the benefit of post-attack fission (decimation/branching) - note that this technique is of no use against volumetric attacks borne by the proxies
- (reactive) adaptive fission so that number of “attacked” servers tends to the number of attackers for maximum quarantine benefit

This research was supported by a DARPA XD3 grant and Amazon AWS Cloud Credits gift. Contact the corresponding author G. Kesidis at gik2@psu.edu
II. BACKGROUND ON STIRLING NUMBERS OF THE SECOND KIND

Suppose $K$ clients attack a system of $M$ servers. So the number of actually attacked servers is $\leq M$.

Generally, the number of ways that $K$ distinct attackers can be assigned to exactly $R \leq K \land M := \min\{K, M\}$ distinct servers (i.e., such that exactly $R$ servers have at least one assigned attacker and $M-R$ servers have no assigned attackers) is

$$\frac{M!}{(M-R)!}\{\frac{K}{R}\} \quad \text{where} \quad \{\frac{K}{R}\} = \frac{1}{R!} \sum_{j=0}^{R} (-1)^{R-j} \binom{R}{j} j^K$$

is a Stirling number of the second kind.

A. A Stirling distribution for distinct attackers and servers

Suppose attackers are assigned to servers in equally likely fashion. This may require that servers have equally-sized regions of key-space assigned by DHT, see Appendix A. If so, the probability that $K$ distinct attackers are assigned to exactly $R$ of $M$ distinct servers is

$$P_K(M, R) := \frac{\binom{M!}{(M-R)!}\{\frac{K}{R}\}}{M^K} \quad \text{where} \quad M^K = \sum_{R=1}^{K\land M} \frac{M!}{(M-R)!}\{\frac{K}{R}\}.$$  

In the following, we refer to the discrete distribution on \{1, 2, ..., $K \land M$\}, $P_K(M, \cdot)$, as a Stirling distribution. Note that the “Stirling distribution of the second kind” (Equation (4.98) of [4]) is different, but see Equation (1.55) of [4].

Moments of a Stirling distribution can be computed by constructing a Markov chain $Y_K \sim P_K(M, \cdot)$ (i.e., $K$ increasing and $M$ fixed). $P(Y_{K+1} = R | Y_K = R) = R/M = 1 - P(Y_{K+1} = R+1 | Y_K = R)$ leading to the following recursion for the $i^{th}$ moment when $K < M$:

$$\begin{align*}
EY_{K+1}^i &= \sum_{R=1}^{K+1} R^i P(Y_{K+1} = R) \\
&= \sum_{R=1}^{K} R^i \frac{R}{M} P(Y_K = R) + \sum_{R=2}^{K+1} R^i (1 - \frac{R-1}{M}) P(Y_K = R-1) \\
&= \frac{1}{M} EY_{K+1}^{i+1} + E(1 + Y_K)^i - \frac{1}{M} E(1 + Y_K)^i
\end{align*}$$

For example, we can solve this recursion for the mean ($i = 1$, noting $Y_1 = 1$ a.s.):

$$\sum_{R=1}^{K} RP_K(M, R) = EY_K = 1 + \frac{M-1}{M} EY_{K-1} = M(1 - (1 - \frac{1}{M})^K) \sim M(1 - e^{-K/M}),$$

which is intuitive since $Y_K = \sum_{m=1}^{M} 1 \{\text{server } m \text{ is not attacked}\}$. Also note that for $K > M$, the limit of summation $EY_{K+1}^i$ is $M$ not $K+1$ and $\lim_{K \to \infty} Y_K = M$ a.s.

We employ the asymptotic approximation 26.8.42 given at [8] to the Stirling numbers of the second kind (the combinatorial quantities involved in computing the Stirling distribution for parameters of interest are immense). Then using logarithms, we computed the distributions $P_K$.

When $K \gg M$ or $K \ll M$, the distribution $P_K(M, \cdot)$ is concentrated at its mode very near $K \land M$. However when $K = M$, the mode (number of attacked servers $\sim P_K(M, \cdot)$) is about 20-25% less than $K = M$, see Figure 1. Figure 2 shows how the mean number of attacked servers $\sum_{R=1}^{K} RP_K(M, R)$ depends on the number initially targeted $M$, where $M$ is in a range close to $K$ (note the saturation to $K$ as $M$ increases).
Fig. 1. $P_K(M, R)$ vs. $R$ for $M = K = 100, 1000, 5000$

Fig. 2. Expected number of attacked servers, $\sum_{R=1}^{K} R P_K(M, R)$, and those free of attack, $M - ER$

B. Generalized $r$-associated Stirling numbers and the probability that $R_n$ servers have at least $n$ attackers

Let $S_{K,n}(M, R_n)$ be the number of ways $K$ distinct attackers can be assigned to $M$ distinct servers so that exactly $R_n \geq 1$ of the servers are assigned at least $n$ attackers. Note that this is not the same as “$r$-associated” Stirling numbers of the second kind, see [1] p. 221,222. Also note that $S_{K,1}(M, R_1) = \left\{\begin{array}{c} K \\ R_1 \end{array}\right\} M!/(M - R_1)!$ for $1 \leq R_1 \leq K \wedge M$.

More generally, let $R_m$ be the number of servers assigned at least $m$ attackers with $R_0 = M$, so $k \leq m$ implies $R_k \geq R_m$. Also let $U_j = \sum_{i=1}^{j} R_j$ and $U_0 = 0$. By first deciding which servers are assigned at least one attacker, and then restricting assignment of the remaining attackers to these servers, we get the following expression

$$S_{K,n}(M, R_n) = \sum_{R_{n+K-nR_n-1}=1}^{K-U_n+K-nR_n-1} \cdots \sum_{R_{n+2}=1}^{K-U_{n+1}} \sum_{R_{n+1}=1}^{K-U_n} \sum_{R_{n-1}=R_n}^{K-R_n-U_n-2} \sum_{R_{n-2}=R_n}^{K-2R_n-U_n-3} \prod_{j=1}^{n+K-nR_n} \frac{(K-U_j-1)!}{(K-U_j-1-R_j)!} \left(\frac{R_{j-1}}{R_j}\right)^{j!R_j-R_{j+1}},$$

where $\sum_{a}^{b}$ is “skipped” if $b < a$ and the final term compensates for permutations within each server, i.e., it does not matter what order the attackers are placed in a server, and $R_{n+K-nR_n+1} = 0$.

Associated probabilities are

$$P_{K,n}(M, R) = S_{K,n}(M, R)/M^K.$$
Again, let $K$ be the number of bots, $U - K$ be the number of benign/nominal users/clients, and $M$ be the number of proxies. Bots will want to have similar session times and network I/O as nominal users so as not to be easily detected and blacklisted.

Given $K$, the number of proxies $R$ currently handling bots has the Stirling distribution $P_K(M, \cdot)$. Let $A$ be distributed as the time required to modify the IP address of a proxy, with $T := EA$. Let $D$ be distributed as the time required to disseminate a proxy’s IP address in the botnet, with $t := ED$. Assume all proxies have back-to-back sessions with some probing bot.

Arguing as for Little’s formula, the average fraction of fresh proxy IP addresses in the botnet is

$$\frac{E(A - D)^+}{EA} \cdot \frac{ER}{M} = \frac{E(A - D)^+}{EA} \left(1 - \left(1 - \frac{1}{M}\right)^K\right)$$

If $A$ and $D$ are modeled as exponentially distributed, then

$$E(A - D)^+/EA = 1 - (1 + T/t)^{-1}.$$  

IV. Reactive Shuffle Quarantine With and Without the Use of Spares for Both Proxies and Replicas

We now consider reacting to an attack targeting a population of servers: either a volumetric attack targeting the proxies or more focused attack targeting the application-server replicas. Again, we assume that the attacked servers cannot determine which particular clients are bots/attackers (otherwise they would be simply blacklisted), that bots can cope with dynamic migration (reassignment) to other servers, i.e., “shuffling” assignments of clients to servers, and that if a server is deemed under to be under attack then no new or known nominal clients are assigned to it.

Again, suppose $K$ clients/bots attack and there are $M$ servers in total. We may reserve $H \geq 0$ servers as hot spares \(^1\) so that, pre-attack, only $M - H$ servers are actively serving clients. After an attack is detected in say $q(0) \geq K \land M - H$ servers, the $H$ spares are added in for the purposes of shuffling. $q(0)$ is an instance of a $P_K(M - H, \cdot)$ distributed random variable. So, the fraction of key-space not affected by the attack is

$$\frac{M - H - q(0)}{M - H}$$

here assuming all of the active $M - H$ servers had a roughly equal portion of key-space, again see Appendix A.

Now suppose the attack-affected portion of key-space (the union of the key-space of the $q(0)$ attacked servers) is randomly and evenly divided among $q(0) + H$ servers, i.e., shuffling the attacked key-space after mobilizing the hot

\(^1\)Spares servers are always ready to be engaged, i.e., “hot”, so that additional delays are not suffered when responding to attack.
where again $q^k$ key-space segments (of the $q$ new identification keys uniformly at random from within the attacked key space and then randomly assigning the $k(q(0) + H)$ keys to the $(q(0) + H)$ servers, $k$ per server. For large $k$, the result is roughly equal-length assignments of key-space to servers, again see the Appendix A.

As a consequent of shuffling, only $q(1) \leq q(0) + H$ of the servers are now assigned an attacking clients, where $q(1) \sim P_K(q(0) + H, \cdot)$. So, the fraction of key-space not affected by the attack has been increased by

$$\frac{q(0)}{M-H} \cdot \frac{q(0) + H - q(1)}{q(0) + H}.$$  \hspace{1cm} (2)

Note that (1) decreases in $H$, but (2) increases with $H$. Thus, there is a potential performance trade-off in the use of spares depending on the system parameters (particularly $K, M$) involved. This trade-off is illustrated in the following numerical examples based the following analysis of expected (mean) performance of shuffling with spares.

Immediately after the attack, the mean fraction of key-space not affected by the attack is

$$L_{K,0}(M - H) := \sum_{q(0)=1}^{K \wedge (M-H)} P_K(M - H, q(0)) \frac{M - H - q(0)}{M - H},$$  \hspace{1cm} (3)

where again $q(0)$ is an instance of the number of affected (attacked) servers immediately after the attack.

Now suppose that the $H \geq 0$ hot spares are introduced and then the attacked key-space is shuffled as described above. The resulting expression for mean fraction of key-space not affected by the attack is obtained by conditioning on $q(0)$:

$$L_{K,1}(M, H) := \sum_{q(0)=1}^{K \wedge (M-H)} P_K(M - H, q(0)) \left( \sum_{q(1)=1}^{K \wedge (q(0)+H)} P_K(q(0) + H, q(1)) \right) \times \left( \frac{M - H - q(0)}{M - H} + \frac{q(0)}{M - H} \cdot \frac{q(0) + H - q(1)}{q(0) + H} \right),$$

where again $q(1)$ is an instance of the number of attacked servers after a single shuffle.

For a simple example, suppose there are $K = 3$ attackers and $M = 20$ servers. If no spares are used, then there could be $q(0) = 3, 2$ or 1 servers attacked. That would respectively correspond to 85%, 90% or 95% of key space not affected. The average $L_{3,0}(20)$ is about 85.7%, see the horizontal line of Figures 4.

In Figures 4-6, three example cases are taken from which we postulate that:

- the use of hot spares $H > 0$ is only beneficial the case where $K \ll M$, i.e.,

$$\forall K \ll M, \quad L_{K,1}(M, H) \quad \text{increases in } H \quad (\text{though sublinearly}).$$

- shuffling is generally beneficial, i.e.,

$$\forall M, \quad L_{K,0}(M) \quad < \quad L_{K,1}(M, 0),$$

see also Figure 1.

Without the aid of spares, subsequent shufflings can occur to further reduce the amount of affected keyspace to $L_{K,i}(M, H)$ for $i \geq 1$, cf., Section V. Each such step requires detection of the subset of servers that continue to be affected by the attack, and balancing the assigned keyspace among (remaining) attacked servers. After a second shuffle, the mean fraction of attacked key-space is

$$L_{K,2}(M, H) := \sum_{q(0)=1}^{K \wedge (M-H)} P_K(M - H, q(0)) \left( \sum_{q(1)=1}^{K \wedge (q(0)+H)} P_K(q(0) + H, q(1)) \sum_{q(2)=1}^{K \wedge q(1)} P_K(q(1), q(2)) \right) \times \left( \frac{q(0)}{M - H} + \frac{q(0)}{M - H} \left( 1 - \frac{q(1)}{q(0) + H} + \frac{q(1)}{q(0) + H} \left( 1 - \frac{q(2)}{q(1)} \right) \right) \right),$$
where \( q(2) \) is an instance of the number of attacked servers after two shuffles and \( H \) spares used only for the first shuffle.

Note that Figures 4-6 suggest using the spares in the first, not subsequent, shuffles, \( i.e. \), when the ratio of the number of attackers to the number of targeted servers is smallest. The system could consist of plural virtual machines each with plural containers where each container acts as a server, in which case the number of “virtual” servers could be very large, \( cf. \), Section VII.

V. REACTIVE SHUFFLES TO ONE-SERVER QUARANTINE (WITHOUT SPARES)

In this section, we explore the benefit of subsequent shuffles among the attacked servers without the use of spares

- nor the use of fission (replica servers), \( cf. \), Section VII,

- nor the consideration of tolerance to more than one, \( cf. \), Section VI.

We define a transient Markov chain \( Q_K(\cdot) \) based on Stirling distributions with a single absorbing state 1, \( i.e. \), \( Q_K(t) \downarrow 1 \) a.s. as the number of shuffles \( t \uparrow \infty \), and compute the expected time (number of shuffles) until the absorbing state is reached and all \( K \) attackers are quarantined in a single server. To connect to the previous model,

\[
Q_K(0) = q(0) + H.
\]

That is, at shuffle (iteration) \( t + 1, t \geq 0 \), and for \( r \in \{1, \ldots, n\} \):

\[
P(Q_K(t + 1) = r \mid Q_K(t) = n) = P_K(n, r)
\]

Using a standard forward-conditioning argument for Markov chains [6], we can find the distribution of the time (number of shuffles) \( T_q \geq 0 \) to quarantine to \( q \) replicas, \( i.e. \), \( T_q \) is the first time \( t \) that \( Q_K(t) = q \). For \( N > q \),

\[
g_{K,q}(N) := \mathbb{E}(T_q | Q_K(0) = N) = \mathbb{E}(\mathbb{E}(T_q | Q_K(1)) | Q_K(0) = N)
\]

\[
= \sum_{r=1}^{N} (g_{K,q}(r) + 1) P_K(N, r)
\]

\[
\Rightarrow g_{K,q}(N) = \sum_{r=1}^{q} P_K(N, r) + \sum_{r=q+1}^{N-1} (g_{K} + 1) P_K(N, r)
\]

\[
\frac{1 - P_K(N, N)}{1 - P_K(N, N)}
\]

with \( g_{K,q}(t) = 0 \) for all \( t \leq q \). Note that a similar forward-equation argument will lead to an expression for the distribution of \( T_1 \), \( i.e. \), \( P(T_1 > t) \).

Generally, the number of attackers is larger than the number of attacked servers, \( i.e. \), \( K \geq N \). In Figures 7-8, we consider \( K = 10 \) and \( K = 15 \) attackers, respectively, on 10 servers.

These graphs confirm intuition that a lot of shuffles are needed to get to a one \( (q = 1) \) quarantine server or a small percentage of the original number of attacked servers. The number of required shuffles grows exponentially fast with the number of attackers \( K \).
The problems with shuffling – i.e., the need to have the number of attackers approximately equal to the number of targeted servers ($K \approx M$) to get 20-25% shuffling gain (see Figure 1) – are mitigated by the ability of proxies to withstand a certain number of (volumetrically) attacking bots or the ability to perform a fission operation on an attacked replica server.

VI. REACTIVE SHUFFLING OF PROXIES - SERVER TOLERANCE OF SOME ATTACKING BOTS

Proxy servers under volumetric DDoS attack can tolerate some attacking bots, recall Section II-B. In particular, recall how Figure 3 shows improved performance (over Figures 1) when a proxy can withstand more than one attacking bot.

VII. REACTIVE FISSION AND CONSOLIDATION OF REPLICA (APPLICATION, WORKER) SERVERS

Now consider a replica (worker, application) server. A single attacker can take down a replica with a localized DoS attack such as Slowloris [9], or targeting a specific protocol supported by the server. Note that the following “fission” technique won’t work for volumetric attacks because the volumetric attack targets the network IO resources of the physical server.

Let $K$ be the number of attackers, $U$ be the number of nominal users, $M$ be the number of application-server replicas, $S(0) \sim P_K(M, \cdot)$ be the number of initially attacked servers.

a) **Fission step:** The total attacked keyspace is divided into $S(0)F$ equal parts (or see Appendix A), where $F$ is the “fission factor.” $F - 1$ replica containers per attacked replica are quickly spun-up or on standby (e.g., $F - 1$ new/standby replica containers in the VM housing the attacked replica). The $S(0)F$ attacked keyspace segments are then assigned to the $S(0)F$ replicas that are the result of $F$-fission of the $S(0)$ attacked replicas, where $F \geq 2$. Collectively, this is called a “fission step” in the following. After the first fission, the number of attacked replicas $S(1) \leq S(0)F$ are identified and the processes repeats.

b) **Fission interleaved with attack-detection and shuffling:** A sequence of fission steps for all detected-attacked replicas may be interleaved with shuffling steps to increase the number of attacked servers to the point where it is on the order of $K$ (an a priori unknown quantity) whereupon a shuffling step will then significantly improve quarantine. One can determine the number of attackers $K$ when the number of container replicas $S$ is reached such that the gain of a subsequent shuffle is 20-24% (so $S \approx K$ by Figure 1). After each attack reassessment, the “liberated” keyspace of replicas not under attack is removed from quarantine and the not-attacked keyspace shared evenly among the not-attacked servers.
c) Summary: In summary, if the attack is very diffuse so that number of initially attacked replicas $S(0) \approx K$, then the first shuffle among attacked replicas will be effective. But if the attack is initially concentrated, i.e., $S(0) \ll K$, then the number of binary $F$-fissions (and ineffective shuffles) until $S \approx K$ (and effective shuffles/fissions commence) is about

$$\log_F(K/S(0))$$

Once the number of attacked servers $S \approx K$ for the first time, the number of subsequent successful shuffles/fissions required to quarantine to 5% of the originally attacked key-space is about

$$\log 0.05/\log 0.78 \approx 12,$$

where $ER \approx .78M$ in Figure 1. Again, though the number of “attacked” replicas is thus maintained to be on the order of $K$, their portion of keyspace is reduced 22% each iteration.

REFERENCES


APPENDIX A: DISTRIBUTION OF RANDOM PARTITIONS IN KEY-SPACE

The following derivations are taken from [7] and added herein for completeness.

Consider $n$ peers whose keys are chosen independently and uniformly at random on a circle of unit circumference and assign to each peer the arc segment for which its key is the left boundary point. Let $A_n$ be a random variable representing lengths of arc segments.

**Lemma 1.**

$$EA_n = \frac{1}{n} \sim \sigma(A_n),$$

where $\sigma$ is standard deviation and the approximation is close for large $n$.

**Proof:** The complementary CDF is

$$P(A_n > x) = (1 - x)^{n-1}.$$ 

One can easily check that

$$EA_n = \int_0^1 P(A_n > x)dx = \frac{1}{n}.$$ 

To compute the standard deviation $\sigma(A_n)$, one can compute the PDF of $A_n$,

$$-\frac{d}{dx}P(A_n > x),$$

and then directly evaluate $EA_n^2$ (integrate by parts twice) and substitute into $\text{var}(A_n) \equiv \sigma^2(A_n) \equiv EA_n^2 - (EA_n)^2$.

Now suppose that each peer has $k \geq 1$ keys assigned, where all $nk$ keys are chosen and arc segments are assigned as for part (a). Let $B_{n,k}$ be a random variable representing the total lengths of the $k$ arc segments of a peer. Clearly, by the previous lemma,

$$EB_{n,k} = kEA_{nk} = \frac{1}{n} = EA_n,$$
i.e., the mean total arc length assigned to a peer does not depend on \( k \).

**Theorem 1.**

\[
\sigma(B_{n,k}) \sim \frac{\sigma(A_n)}{\sqrt{k}}.
\]

**Proof:**

Let \( N = nk \) and pick \( k \) distinct indices from \( \{1, \ldots, N\} \) at random, independently of the original experiment, and let \( B \) be the sum of the lengths corresponding to these indices. More precisely, let \((\xi_1, \ldots, \xi_k)\) be a sampling without replacement from \( \{1, \ldots, nk\} \) and let

\[
B = A_{\xi_1} + \cdots + A_{\xi_k}.
\]

Observation 1: The random vector \((A_1, \ldots, A_N)\) is permutable, and so is \((\xi_1, \ldots, \xi_k)\). Therefore

\[
B \sim A_1 + \cdots + A_k.
\]

Observation 2: Let \( X_1, X_2, \ldots \) be i.i.d. exponentials with mean 1, and let \( S_n = X_1 + \cdots + X_n \). Then

\[
(A_1, \ldots, A_N) \sim (X_1, \ldots, X_N | S_N = 1)
\]

by conditional uniformity. From this you can directly compute \( \mathbb{E}(A_1 + \cdots + A_k | S_N) \) and \( \text{var}(A_1 + \cdots + A_k | S_N) \) and then take the values at \( S_N = 1 \). To compute the density \( f_B \) of \( B \), note that

\[
B \sim (S_k | S_N = 1),
\]

implies

\[
P(B \in dx) = \frac{P(S_k \in dx, S_N \in dt)}{P(S_N \in dt)} \bigg|_{t=1}.
\]

Let \( f_m(x) \) be the density of \( S_m \) (which is Gamma):

\[
f_m(x) = \frac{x^{m-1}}{(m-1)!} e^{-x}.
\]

Hence

\[
f_B(x) = \frac{f_k(x) f_{N-k}(1-x)}{f_N(1)} \propto x^{k-1}(1-x)^{(n-1)k-1}
\]

which is Beta with parameters

\[
\alpha = k, \quad \beta = (n - 1)k.
\]

The Beta distribution has the property that

\[
\mathbb{E}B = \frac{\alpha}{\alpha + \beta}, \quad \text{var}(B) = \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}.
\]

So \( \mathbb{E}B = k/N = 1/n \), and

\[
\text{var}B = \frac{n - 1}{n^2 (nk + 1)} \sim \frac{1}{n^2 k} \quad \text{as } n \to \infty.
\]

\( k = O(\log n) \) was suggested for Chord [5].
APPENDIX B: THE CASE WHERE $R_n$ SERVERS HAVE AT LEAST $n$ INDISTINCT ATTACKERS

We can count the number of ways that a certain number $R_n$ of servers are each assigned at least $n$ attackers by assuming that the attackers are indistinct while the servers are distinct, as in Bose-Einstein statistics.

First consider the number of ways that $M$ whole numbers (each representing a server containing that number of attackers) can sum to $K$ where the order of the sum matters (i.e., distinct servers). To compute this, list $K$’s and $M−1$ other identical symbols, say *, in a row. The number of 1’s to the left of the leftmost * is the first (leftmost) whole number, that between the $i^{th}$ and $(i+1)^{st}$ * is the $(i+1)^{st}$ whole number, while that to the right of the rightmost * is last ($M^{th}$) whole number. Since the symbols 1 and * are identical, they can be permuted without changing the instance. So, the answer is by choice with replacement:

$$\binom{K + M - 1}{M - 1}.$$ 

Let $V_{K,n}(M, R_n)$ be the number of ways $K$ indistinct attackers can be assigned to $M$ distinct servers so that exactly $R_n \geq 1$ of the servers are assigned at least $n$ attackers. Consider an arbitrary such instance. For $m \leq n$, let $R_m$ be the number of servers in the instance that are assigned at least $m$ attackers, so that $k \leq m$ implies $R_k \geq R_m$. By first deciding which servers are assigned at least one attacker, and then restricting assignment of the remaining attackers to these servers, we get

$$V_{K,1}(M, R_1) = \binom{M \setminus R_1}{R_1} \left(\frac{(K - R_1) + (R_1 - 1)}{R_1 - 1}\right) \text{ where } 1 \leq R_1 \leq M \wedge K.$$ 

For $V_{K,2}(M, R_2)$, we can decide on the servers with at least 1 attacker, then on the servers with at least 2:

$$V_{K,2}(M, R_2) = \sum_{R_1=R_2}^{(K-R_2)\wedge M} \binom{M}{R_1} \left(\frac{(K - R_1 - R_2) + (R_2 - 1)}{R_2 - 1}\right)$$

Letting $R_0 = M$, $U_j = \sum_{i=1}^{j} R_j$ and $U_0 = 0$, we can show by induction that

$$V_{K,n}(M, R_n) = \sum_{R_{n-1}=R_n}^{(K-R_n-U_{n-2})\wedge R_{n-2}} \sum_{R_{n-2}=R_n}^{(K-2R_n-U_{n-3})\wedge R_{n-3}} \cdots \sum_{R_2=R_n}^{(K-(n-2)R_n-U_1)\wedge R_1} \sum_{R_1=R_n}^{(K-(n-1)R_n-U_0)\wedge R_0} \binom{K - U_n + (R_n - 1)}{R_n - 1} \prod_{j=1}^{n} \binom{R_{j-1}}{R_j},$$

where $\sum_{a}^{b}$ is “skipped” if $b < a$.

But each instance thus counted is not “equally likely,” so that the probability that exactly $n$ servers have at least $R_n$ attackers is not $V_{K,n}(M, R_n)/(\binom{K+M-1}{M-1})$. For example, take $M = 4$ and $K = 5$ and $n = 2$ (and, again, servers are always distinct). With indistinct attackers, there are four ways that servers 2 and 3 have at least two attackers each:

*111*11  *11*1111  1*11*11  *11*111

But with distinct attackers, the first two correspond to $\binom{5}{3} = 10$ ways each, while the last two correspond to $\binom{5}{2} = 10$ ways each.