Privacy Leakage in Discrete-Time Updating Systems

Nitya Sathyavageeswaran, Roy D. Yates, Anand D. Sarwate, Narayan Mandayam

Department of Electrical and Computer Engineering, Rutgers University

E-mail: {nitya.s, anand.sarwate}@rutgers.edu, {ryates, narayan}@winlab.rutgers.edu

Abstract—**A source generates time-stamped update packets that are sent to a server and then forwarded to a monitor. This occurs in the presence of an adversary that can infer information about the source by observing the output process of the server. The server wishes to release updates in a timely way to the monitor but also wishes to minimize the information leaked to the adversary. We analyze the trade-off between the age of information (AoI) and the maximal leakage for systems in which the source generates updates as a Bernoulli process. For a time slotted system in which sending an update requires one slot, we consider three server policies: (1) Memoryless with Bernoulli Thinning (MBT): arriving updates are queued with some probability and head-of-line update is released after a geometric holding time; (2) Deterministic Accumulate-and-Dump (DAD): the most recently generated update (if any) is released after a fixed time; (3) Random Accumulate-and-Dump (RAD): the most recently generated update (if any) is released after a geometric waiting time. We show that for the same maximal leakage rate, the DAD policy achieves lower age compared to the other two policies but is restricted to discrete age-leakage operating points.**

Index Terms—**Age of information, maximal leakage, status updates, Bernoulli process.**

I. INTRODUCTION

A smart home has a range of devices that can be accessed or controlled remotely using the internet. The various sensors in a smart home send time-stamped updates about the temperature, humidity, power consumption, etc. to a destination monitor. In spite of the advantages offered by a smart home system, there is also a potential loss of privacy. Adversaries could infer sensitive information about the home occupants from the update packets generated from the various sensors. Moreover, timeliness of the updates is important in many applications, but improving the timeliness of delivered updates can increase the information leaked to the adversary. When the timeliness of these updates is important, an age of information metric is useful in optimizing these systems. The age of information metric tells us how much time has elapsed since the generation of the most recent update that has been received at the monitor [1].

In this paper we study trade-offs between privacy and age. We consider a model for a smart home system in which a source generates time-stamped updates that are sent to a monitor through a server. There is an adversary present at the monitor that can infer information about the source from the packet arrival process. We measure privacy using the maximal leakage metric introduced by Issa, Kamath, and Wagner [2]; this measures the maximal multiplicative gain that the adversary can guess any function of the original data from

the observed data. The maximal leakage increases when we minimize age in these systems. We design service policies in order to reduce the maximal leakage and see the effect on the age of information (AoI) for these systems.

The time average AoI for various systems has been extensively studied, including the first-come first-served (FCFS) M/M/1, M/D/1 and D/M/1 queues, and the last-come first-served (LCFS) queues [1], [3]–[9]. AoI has also been analyzed for other continuous time queueing disciplines [10]–[17] and for various discrete time queues [18]–[21]. Minimizing the timing information leaked, while keeping the status updates timely in an energy harvesting channel is analyzed in [22].

Minimizing the maximal leakage subject to a cost constraint has been studied in [23]–[26]. Issa et al. [27] derive the maximal leakage for both an M/M/1 queue and an "accumulate-and-dump" system.

Our contribution is to study the trade-off between maximal leakage and the age. We examine the discrete time analogues of the M/M/1 queue and the "accumulate-and-dump" system. We also introduce a "Random Accumulate-and-Dump" service policy. We derive the age and maximal leakage for these systems. This raises interesting questions about the general problem of balancing timeliness and privacy in communication systems.

II. SYSTEM MODEL AND PAPER OVERVIEW

Our discrete time system consists of a source, server, monitor and an adversary as shown in Figure 1. Time passes in integer slots with slot $n \geq 0$ denoting the time interval $[n, n+1)$. The transmission of a packet requires one slot. A packet sent in slot n (from source to server or from server to monitor) is received at the start of slot $n+1$.

A. Information Leakage

To characterize information leakage, packet transmissions from the source to the server are indicated by the binary sequence $X^n = (X_1, \ldots, X_n)$ such that $X_t = 1$ if the server receives a packet from the source at the start of slot t. This packet will have been generated and transmitted by the source in slot $t - 1$. Packet transmissions from the server to the monitor are similarly indicated by the binary sequence $Y^n = (Y_1, \ldots, Y_n)$ such that $Y_t = 1$ if the server sends a packet in slot t. The adversary observes the server transmission sequence Y^n. The updating policy of the source, coupled with the updating policy of the server induces a joint PMF $P_{X^n, Y^n}(x^n, y^n)$. From this PMF, the support set of X^n is denoted $\mathcal{X}_n = \{x^n \colon P_{X^n}(x^n) > 0\}$. The support set $\mathcal{Y}_n$ of

Fig. 1. A source sends status updates through a server to a monitor. An adversary (Adv) observes transmissions from the server to the monitor.

Y^n is defined analogously. We measure the information leaked to the adversary by the maximal leakage metric.

Definition 1 (Issa et al. [27]). *A joint distribution $P_{X^n Y^n}$ on finite alphabets $\mathcal{X}^n$ and $\mathcal{Y}^n$ has maximal leakage*

$$\mathcal{L}(X^n \to Y^n) = \log \sum_{y^n \in \mathcal{Y}_n} \max_{x^n \in \mathcal{X}_n} P_{Y^n|X^n}(y^n|x^n). \quad (1)$$

The key technical challenge in this work is to find the leakage maximizing input sequence $x^n \in \mathcal{X}_n$ for each possible output sequence $Y^n = y^n$. Note that the maximal leakage depends on the arrival process X^n only through its support $\mathcal{X}_n$. We find the leakage maximizing input x^n for *full support* arrival processes satisfying $\mathcal{X}_n = \{0, 1\}^n$. We further assume the source sends fresh updates as a rate λ Bernoulli process since this is the simplest class of arrival processes with full support.

B. Age of Information

For the evaluation of update timeliness, we employ a discrete-time age process model that is consistent with prior work [28]. The source generates time-stamped update packets (or simply updates) that are sent to the server. An update generated by the source in slot t is based on a measurement of a process of interest that is taken at the beginning of the slot and has time-stamp t. At the end of slot t, or equivalently at the start of slot $t + 1$, that packet will have age 1. In slot $t + j$, this update will have age j. We say one packet is fresher than another if its age is smaller.

An observer of these updates measures timeliness by a discrete-time *age* process $A(t)$ that is constant over a slot and equals the age of the freshest packet received prior to slot t. When $u(t)$ denotes the time-stamp of the freshest packet observed/received prior to slot t, the age in slot t is $A(t) = t - u(t)$. We characterize the timeliness performance of the system by the average age of information (AoI) at the monitor.

Definition 2 (Kaul et al. [1]). *A stationary ergodic age process has age of information (AoI)*

$$\mathrm{E}[A(t)] = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} A(t), \quad (2)$$

where $\mathrm{E}[\cdot]$ *is the expectation operator.*

C. Server Policies

The server wishes to balance the timing information leaked to the adversary against the AoI at the monitor. To characterize these conflicting objectives, we explore age-leakage trade-offs for three server policies:

- **Memoryless with Bernoulli thinning (MBT)**: The server admits each arriving update into a first-come first-served queue with probability α. If the queue is not empty at the start of a slot, the update at the head of the queue is sent to the monitor with probability μ, independent of transmissions in other slots.
- **Deterministic Accumulate-and-Dump (DAD)**: The server stores the freshest update received from the source. For a fixed parameter τ, the server sends its stored update every τ slots.
- **Random Accumulate-and-Dump (RAD)**: The server stores the freshest update received from the source. In each slot t, the server sends its stored update with probability μ, independent of transmissions in other slots.

With rate λ Bernoulli arrivals, the MBT server acts as a Geo/Geo/1 queue with effective arrival rate $\alpha\lambda$. Furthermore, while the service time of an update is always one slot, each update spends a geometric number of slots at the head-of-line (HOL) position of the queue and the departure process is indistinguishable from that of a discrete-time Geo/Geo/1 queue. It is the discrete-time version of the M/M/1 queue.

The DAD policy was introduced Issa et al. [27] with the generalization that up to n packets are dumped at a time. Here, we focus on the special case of DAD in which no more than $n = 1$ packet is "accumulated" and dumped because maximal leakage grows linearly with n [27] and because the AoI is the same for all $n \geq 1$ as long as the dumped updates include the most recent update. For the same reasons, the RAD server also accumulates only one packet. We further note that with Bernoulli arrivals, RAD is the discrete-time analogue of the M/M/1/1 queue with preemption in service; the time an update spends in the HOL position is geometric and in each time slot the HOL update may be replaced by a new arrival.

III. MAXIMAL LEAKAGE

A fixed arrival sequence x^n and service policy induces a conditional PMF $P_{Y^n|X^n}(y^n|x^n)$. For the MBT and RAD servers, the service time of a packet is geometric with parameter μ. For these servers, we employ the following lemma to calculate the maximal leakage.

Lemma 1. *For the MBT and RAD servers with full support arrival processes,*

$$\max_{x \in \mathcal{X}^n} P_{Y^n|X^n}(y^n|x^n) = \mu^{\sum_{i=1}^n y_i} \quad (3)$$

and this maximum is achieved when $x^n = y^n$.

The proof follows by an induction argument; details are in the extended version of this manuscript. Lemma 1 shows that for a given output sequence y^n from either the MBT or the

RAD server, the maximal leakage is maximized by a "just-in-time" input $x^n = y^n$ in which the source generates each update in the slot just prior to its departure from the server. The MBT and the RAD server share the property that the HOL update in each slot is transmitted with probability μ. The just-in-time input $x^n = y^n$ maximizes $P_{Y^n|X^n}(y^n|x^n)$ because it precludes departures from the server prior to the departures specified by the output y^n.

Theorem 1. *For full support arrival processes, the maximal leakage rate for the MBT and RAD servers is given by*

$$\frac{1}{n}\mathcal{L}(X^n \to Y^n) = \log(1+\mu). \tag{4}$$

Proof. Applying Lemma 1 to (1) yields

$$\mathcal{L}(X^n \to Y^n) = \log \sum_{y^n \in \mathcal{Y}^n} \mu^{\sum_{i=1}^n y_i}$$
$$= \log \sum_{y_1=0}^1 \mu^{y_1} \cdots \sum_{y_n=0}^1 \mu^{y_n}$$
$$= \log \left(\mu^0 + \mu^1\right)^n = n\log(1+\mu). \tag{5}$$

$\square$

For the DAD policy, the most recently generated packet is dumped after every τ slots and Y^n is a deterministic function of X^n. Defining $K = \lfloor n/\tau \rfloor$, Y^n has the form

$$Y^n = (0^{\tau-1}, Y_\tau, 0^{\tau-1}, Y_{2\tau}, \ldots, Y_{K\tau}, 0^{n-K\tau}) \tag{6}$$

where $Y_{k\tau} = 0$ if and only if $(X_{(k-1)\tau+1}, \cdots, \mathcal{X}_{k\tau}) = 0^\tau$ and otherwise $Y_{k\tau} = 1$. This structure simplifies the leakage calculation.

Lemma 2. *For the DAD policy,* $\max_{x \in \mathcal{X}^n} P_{Y^n|X^n}(y^n|x^n) = 1$ *and the maximum is achieved when* $x^n = y^n$.

Proof. For a given output $Y^n = y^n$ from the DAD server, the input $x^n = y^n$ implies $P_{Y^n|X^n}(y^n|x^n) = 1$. This is the maximizing input since it achieves the unity upper bound. $\square$

We note that the maximizing x^n may not be unique.

Theorem 2. *The DAD policy has maximal leakage rate*

$$\frac{\mathcal{L}(X^n \to Y^n)}{n} = \frac{\log 2}{n}\left\lfloor \frac{n}{\tau} \right\rfloor. \tag{7}$$

Proof. Since Y^n has the form (6), the support set $\mathcal{Y}_n$ has size $|\mathcal{Y}_n| = 2^K = 2^{\lfloor n/\tau \rfloor}$. Applying Lemma 2 to (1),

$$\mathcal{L}(X^n \to Y^n) = \log \sum_{y^n \in \mathcal{Y}_n} 1 = \log 2^{\lfloor n/\tau \rfloor} = \left\lfloor \frac{n}{\tau} \right\rfloor \log 2. \tag{8}$$

$\square$

Since $\lim_{n\to\infty} \lfloor n/\tau \rfloor / n = 1/\tau$, it follows from Theorem 2 that the DAD policy has maximal leakage rate

$$\lim_{n\to\infty} \frac{\mathcal{L}(X^n \to Y^n)}{n} = \frac{\log 2}{\tau}. \tag{9}$$

Fig. 2. The source sends fresh updates to the server in slots $N_k = n_k$, inducing the age process $A_i(n)$ at the input to the server. The server sends samples of the most recent update to the monitor in time slots $S_k = s_k$, inducing the age process $A_m(n)$ at the monitor.

IV. AGE OF INFORMATION

The MBT server with rate λ Bernoulli arrivals and admission probability α is identical to a Geo/Geo/1 queue with arrival rate $\alpha\lambda$. If the server queue is not empty at the start of slot t, the server sends the head-of-line update with probability μ. Using the discrete time analogue of the M/M/1 graphical AoI analysis [1], the AoI of the Geo/Geo/1 queue has been previously derived by Kosta et al. [28] and equals

$$\mathrm{E}[A_{\mathrm{MBT}}] = \frac{1}{\alpha\lambda} + \frac{1-\alpha\lambda}{\mu-\alpha\lambda} - \frac{\alpha\lambda}{\mu^2} + \frac{\alpha\lambda}{\mu}. \tag{10}$$

While the graphical method could be employed for age analysis of the DAD and RAD systems, here we instead employ the sampling of age processes approach [10]. In the implementation of the DAD/RAD servers, no update is dumped if no update arrived in the preceding inter-dump period. However, for the purpose of AoI analysis, we make a different assumption: if no update arrives before the dump attempt, then the server resends its previously dumped update. This approach is an example of the method of fake updates introduced in Yates and Kaul [29]. With respect to the age at the monitor, in the absence of an update to dump, it makes no difference if the server sits idle or if the server repeats sending the prior update. Neither changes the age at the monitor.

However this fake update approach simplifies the age analysis. We can now view the monitor as *sampling* the update process of the server. With each dump instance, the monitor receives the freshest update of the server, resetting the age at monitor to the age of the dumped update. In the same way, we can view the server as sampling the always-fresh update process of the source.

For both the DAD and RAD systems, we observe that the source offers fresh updates to the server as a renewal process. The dump attempts of the DAD and RAD servers are also renewal processes. Specifically, the DAD server attempts to release its freshest update every τ slots while the RAD server has dump attempts with iid geometric inter-dump times.

A. Sampling the Source: The Age of Fresh Updates

Referring to Figure 1, the source can generate a fresh (age zero) update in a slot n and forward it to the server in that same slot. This fresh update arrives at the server at the end of slot n with age 1. The age process $A_i(n) = Z_n$ of an observer at the server input is reset at the start of slot $n + 1$ to $Z_{n+1} = 1$. When a source generates fresh updates in slots $N_k = n_k$, the age Z_n evolves as the sequence of staircases shown in Figure 2. From the observer's perspective, update k *arrives* in slot $N_k + 1$ with interarrival time

$$Y_k = N_k + 1 - (N_{k-1} + 1) = N_k - N_{k-1}. \quad (11)$$

Defining the indicator $I_{\{A\}}$ to be 1 if event A occurs and zero otherwise, we can employ Palm probabilities to calculate the age PMF

$$P[Z_n = z] = \lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} I_{\{Z_n = z\}}. \quad (12)$$

The sum on the right side of (12) can be accumulated as rewards over each renewal period. In the kth renewal period, we set the reward R_k equal to the number of slots in the kth renewal period in which $Z_n = z$. If $Y_k \geq z$, then there is one slot in the renewal period in which $Z_n = z$ and thus the reward is $R_k = 1$; otherwise $R_k = 0$. Thus, for $z = 1, 2, \ldots$, $R_k = I_{\{Y_k \geq z\}}$. From renewal reward theory, it follows from (12) that

$$P[Z_n = z] = \frac{E[R_k]}{E[Y_k]} = \frac{P[Y_k \geq z]}{E[Y_k]}, \quad z = 1, 2, \ldots. \quad (13)$$

This is the discrete-time version of a well-known distribution of the age of a renewal process. It follows from (13) that the average age of the current update at the server has average age

$$E[A_i(n)] = E[Z_n] = \sum_{z=1}^{\infty} z\, P[Z_n = z] = \frac{E[Y_k^2]}{2\,E[Y_k]} + \frac{1}{2}. \quad (14)$$

B. Sampling the Server: The Age at the Monitor

Now we examine age at the monitor for the DAD and RAD servers. As depicted in Figure 2, the server (whether DAD or RAD) sends its freshest update to the monitor at sample times $S_1, S_2, \ldots$. These sample times S_k also form a renewal process with iid inter-sample times $Y'_k = S_k - S_{k-1}$. For the DAD server, $Y'_k = \tau$. while for the RAD server, the Y'_k are geometric (μ) random variables. We now analyze the average age at the monitor in terms of the moments of Y'_k.

The age process $A_i(n)$ at the input to the server is identical to the Z_n process defined in Section IV-A. When the server sends its most recent update to the monitor in slot S_k, this update has age $A_i(S_k)$ at the start of the slot. The monitor receives the update at the end of the slot with age $A_i(s_k) + 1$. Thus, at the start of slot $s_k + 1$, the age at the monitor is reset to $A_m(s_k + 1) = A_i(s_k) + 1$. Graphically, this is depicted in Figure 2.

To describe the monitor age $A_m(n)$ in an arbitrary slot n, we look backwards in time and define Z'_n as the age of the renewal process defined by the inter-renewal times Y'_k associated with sampling the server. In Fig. 2 for example, in slot $n = 11$, the last update was sampled by the server at time $s_2 = 7$ and $Z'_{11} = 11 - 7 = 4$. We note that the Z'_n process is the same as the Z_n process, modulo the inter-renewal times now being labeled Y'_k rather than Y_k. In particular the PMF $P[Z'_n = z]$ and expected value $E[Z'_n]$ are described by (13) and (14) with Y'_k replacing Y_k.

The age at the monitor in slot n is then

$$A_m(n) = A_i(n - Z'_n) + Z'_n. \quad (15)$$

When the age process $A_i(n)$ at the input to the monitor is stationary and the sampling process that induces Z'_n is independent of the $A_i(n)$ age process, it follows that $A_i(n)$ and $A_i(n - Z'_n)$ are identically distributed. Thus, the average age at the monitor is

$$
\begin{aligned}
E[A_m(n)] &= E[A_i(n - Z'_n)] + E[Z'_n] \\
&= E[A_i(n)] + E[Z'_n]. \quad (16)
\end{aligned}
$$

Employing (14) to evaluate both $E[A_i(n)]$ and $E[Z'_n]$, we obtain

$$E[A_m(n)] = \frac{E[Y_k^2]}{2\,E[Y_k]} + \frac{1}{2} + \frac{E[(Y'_k)^2]}{2\,E[Y'_k]} + \frac{1}{2}. \quad (17)$$

For both the DAD and RAD servers, the source generates packets as a rate λ Bernoulli process with $E[Y_k] = 1/\lambda$ and $E[Y_k^2] = (2 - \lambda)/\lambda^2$. It follows from (14) that $E[A_i(n)] = 1/\lambda$. For the DAD server, $Y'_k = \tau$ is deterministic and (14) implies $E[Z'_n] = (\tau + 1)/2$. For the RAD server, the inter-sample times Y'_k at the monitor are geometric (μ) and $E[Z'_n] = 1/\mu$. With these observations, we have the following theorem.

Theorem 3. *When the source emits updates as a rate λ Bernoulli process, the average age at the monitor is*

$$E[A_m(n)] = \begin{cases} 1/\lambda + (\tau + 1)/2, & \text{DAD server,} \\ 1/\lambda + 1/\mu, & \text{RAD server.} \end{cases} \quad (18)$$

V. MAXIMAL LEAKAGE VS AVERAGE AOI

Fig. 3 shows the variation of age with maximal leakage rate for MBT with $\alpha = 1$ (which is Geo/Geo/1), DAD, and RAD systems with arrival rate $\lambda = 0.5$. As the service rate for these systems increases, the maximal leakage rate for the three service policies increases and the age decreases. Note that for the service rate of 1 ($\mu = 1$ or $\tau = 1$), the three systems have identical service service processes and thus the same average age of 3 slots.

At other values of the service rate, the DAD system has better age than the other two systems for the same maximal leakage rate. The DAD system keeps the age small by not queueing packets. It delivers only the most recent update packet. From Theorem 3, we see for the DAD and RAD systems that for a fixed value of the maximal leakage rate (as specified by τ or μ) the age is decreasing in the source rate λ and this is illustrated in Fig. 4.

In Fig. 5 we show the variation of the age with the maximal leakage rate for the MBT server for various arrival rates λ.

Fig. 3. The age vs. maximal leakage rate for the MBT ($\alpha = 1$), DAD, and RAD servers with arrival rate $\lambda = 0.5$. The service rate μ varies over $[0.524, 1]$ for Geo/Geo/1 and over $[0.05, 1]$ for RAD. For DAD, τ varies from 1 to 39.

Fig. 4. The age vs. maximal leakage rate for the RAD and DAD policies for $\lambda = 0.1$, $\lambda = 0.5$ and $\lambda = 0.9$.

With fixed λ and $\alpha = 1$, the server controls the leakage via the service rate μ. As μ is reduced and approaches λ, the maximal leakage is reduced but the age blows up as the queue backlog increases. In fact, we can achieve better trade-offs by varying both μ and the admission probability α. For a given λ, we can choose α to operate at an effective arrival rate $\alpha\lambda < \lambda$. This stabilizes the queue by throwing away arrivals to keep the backlog small. With λ fixed, for each μ (which specifies the leakage), the $\alpha \in (0, 1]$ that minimizes $\mathrm{E}[A_{\mathrm{MBT}}]$ in (10) is calculated. This is shown in Fig. 5 by the solid blue-orange-green "$\alpha\lambda$" trade-off curve. The blue segment is achievable for $\lambda \geq 0.1$, the orange segment is achievable for $\lambda \geq 0.5$ and the green segment is achievable for $\lambda \geq 0.9$.

VI. CONCLUSIONS AND FUTURE WORK

In this work we examined the trade-off between the maximal leakage and the average age for three service policies. For a given arrival rate λ, the DAD server is constrained by integer

Fig. 5. The age vs. maximal leakage rate for the MBT system: for $\alpha = 1$, the $\lambda = 0.1$, $\lambda = 0.5$ and $\lambda = 0.9$ age-leakage trade-off curves are obtained by varying μ over the interval $[\lambda + \epsilon, 1]$. With λ fixed, for each μ (which specifies the leakage), the $\alpha \in (0, 1]$ that minimizes $\mathrm{E}[A_{\mathrm{MBT}}]$ in (10) is also calculated. This yields the solid blue-orange-green trade-off curve. The blue segment shows $\alpha\lambda \in [0.054, 0.1]$ and is achievable for $\lambda \geq 0.1$. The orange segment shows $\alpha\lambda \in [0.1, 0.5]$ and the green segment shows $\alpha\lambda \in [0.5, 0.9]$; these segments are achievable for $\lambda \geq 0.5$ and $\lambda \geq 0.9$ respectively.

values of τ to certain age-leakage operating points. For the same maximal leakage, neither the RAD server nor the MBT server could achieve the same average age. However, the MBT and the RAD servers have the flexibility to choose the service rate μ continuously. In addition, the MBT server can choose to thin the arrival process by operating at an effective arrival rate $\alpha\lambda$ to achieve better age-leakage trade-off.

Our initial results here suggest several interesting avenues for future investigation. We limited our analysis to Bernoulli arrival processes that simplify the leakage analysis because all input sequences x^n are in the support set. Extending the analysis beyond Bernoulli arrival processes could change which policies have more favorable trade-offs. Other arrival processes may not have support over all possible input sequences x^n which may make the calculation of maximal leakage more challenging. Likewise, we have limited our attention to specific server policies. More generally, we would like to find "optimal" policies (within a given class) to manage the age-leakage trade-off for a given arrival process or arrival rate. Finally, understanding these trade-offs in networks of status updating systems might open up new avenues for modeling the interplay between delay and privacy.

REFERENCES

[1] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2731–2735.

[2] I. Issa, S. Kamath, and A. B. Wagner, "An operational measure of information leakage," in *2016 Annual Conference on Information Science and Systems (CISS)*, 2016, pp. 234–239.

[3] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, "Effect of message transmission path diversity on status age," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1360–1374, 2016.

[4] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 1583–1587.

[5] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2012, pp. 1–6.

[6] E. Najm and R. Nasser, "Age of information: The gamma awakening," in *2016 IEEE International Symposium on Information Theory (ISIT)*. Ieee, 2016, pp. 2574–2578.

[7] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2569–2573.

[8] R. D. Yates, "Age of information in a network of preemptive servers," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 118–123.

[9] E. Najm and E. Telatar, "Status updates in a multi-stream M/G/1/1 preemptive queue," in *IEEE Infocom 2018-Ieee Conference On Computer Communications Workshops (Infocom Wkshps)*. IEEE, 2018, pp. 124–129.

[10] R. D. Yates, "The age of information in networks: Moments, distributions, and sampling," *IEEE Transactions on Information Theory*, vol. 66, pp. 5712–5728, 2020.

[11] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "The stationary distribution of the age of information in FCFS single-server queues," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 571–575.

[12] A. Soysal and S. Ulukus, "Age of information in G/G/1/1 systems," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 2022–2027.

[13] R. D. Yates, "Status updates through networks of parallel servers," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 2281–2285.

[14] R. Talak, S. Karaman, and E. Modiano, "When a heavy tailed service minimizes age of information," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 345–349.

[15] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Controlling the age of information: Buffer size, deadline, and packet replacement," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 301–306.

[16] ——, "On the age of information with packet deadlines," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6419–6428, 2018.

[17] Y. Inoue, "Analysis of the age of information with packet deadline and infinite buffer capacity," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2639–2643.

[18] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Non-linear age of information in a discrete time queue: Stationary distribution and average performance analysis," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[19] V. Tripathi, R. Talak, and E. Modiano, "Age of information for discrete time queues," *arXiv preprint arXiv:1901.10463*, 2019.

[20] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The age of information in a discrete time queue: Stationary distribution and non-linear age mean analysis," *IEEE Journal on Selected Areas in Communications*, vol. PP, pp. 1–1, 03 2021.

[21] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 15–28, 2019.

[22] O. Ozel, A. Yener, and S. Ulukus, "State amplification and masking while timely updating," *arXiv preprint arXiv:2202.11682*, 2022.

[23] B. Wu, A. B. Wagner, and G. E. Suh, "Optimal mechanisms under maximal leakage," in *2020 IEEE Conference on Communications and Network Security (CNS)*, 2020, pp. 1–6.

[24] J. Liao, O. Kosut, L. Sankar, and F. P. Calmon, "Privacy under hard distortion constraints," in *2018 IEEE Information Theory Workshop (ITW)*, 2018, pp. 1–5.

[25] J. Liao, L. Sankar, F. P. Calmon, and V. Y. F. Tan, "Hypothesis testing under maximal leakage privacy constraints," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 779–783.

[26] I. Issa, S. Kamath, and A. B. Wagner, "Maximal leakage minimization for the Shannon cipher system," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 520–524.

[27] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1625–1657, 2020.

[28] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Queue management for age sensitive status updates," *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 330–334, 2019.

[29] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *2012 IEEE International Symposium on Information Theory Proceedings*, 2012, pp. 2666–2670.

VII. Appendix

Proof of Lemma 1

We prove this claim by induction. We start by showing the claim is true for $n = 1$. For $n = 1$,

$$\mathbb{P}(Y_1 = 1|X_1 = 1) = \mu, \quad \mathbb{P}(Y_1 = 1|X_1 = 0) = 0. \quad (19)$$

It follows from (19) that

$$\max_{x \in \mathcal{X}} P(Y_1 = 1|X_1 = x) = \mu. \quad (20)$$

Similarly,

$$\mathbb{P}(Y_1 = 0|X_1 = 1) = 1 - \mu, \quad \mathbb{P}(Y_1 = 0|X_1 = 0) = 1. \quad (21)$$

It follows from (21) that

$$\max_{x \in \mathcal{X}} P(Y_1 = 0|X_1 = x) = 1. \quad (22)$$

From (20) and (22) we can say,

$$\max_{x \in \mathcal{X}} P_{Y_1|X_1}(y_1|x) = \mu^{y_1}, \quad (23)$$

and the maximum is achieved when $x_1 = y_1$. Thus the claim holds for $n = 1$.

Now let us say the claim holds for some $n = k$. Then,

$$\max_{x^k \in \mathcal{X}^k} P_{Y^k|X^k}(y^k|x^k) = \mu^{\sum_{i=1}^k y_i}, \quad (24)$$

and this maximum is achieved when $x^k = y^k$.

Now we need to show the claim holds for $n = k + 1$.

$$P(y^{k+1}|x^{k+1}) = P(y^k|x^{k+1})P(y_{k+1}|y^k, x^{k+1}). \quad (25)$$

Since the departure process until time k does not depend on the arrival at time $k + 1$, we can write

$$P_{Y^k|X^{k+1}}(y^k|x^{k+1}) = P_{Y^k|X^k}(y^k|x^k). \quad (26)$$

Hence,

$$
\begin{aligned}
&\max_{x^{k+1} \in \mathcal{X}^{k+1}} P_{Y^{k+1}|X^{k+1}}(y^{k+1}|x^{k+1}) \\
&= \max_{x^{k+1} \in \mathcal{X}^{k+1}} P(y^k|x^k)P(y_{k+1}|y^k, x^{k+1}) \\
&\leq [\max_{\hat{x}^{k+1} \in \mathcal{X}^{k+1}} P(y^k|\hat{x}^k)][\max_{x^{k+1} \in \mathcal{X}^{k+1}} P(y_{k+1}|y^k, x^{k+1})] \\
&= [\max_{\hat{x}^k \in \mathcal{X}^k} P(y^k|\hat{x}^k)][\max_{x^{k+1} \in \mathcal{X}^{k+1}} P(y_{k+1}|y^k, x^{k+1})]. \quad (27)
\end{aligned}
$$

By the induction hypothesis,

$$\max_{\hat{x}^k \in \mathcal{X}^k} P_{Y^k|X^k}(y^k|\hat{x}^k) = \mu^{\sum_{i=1}^k y_i}, \quad (28)$$

and the maximum is achieved when $\hat{x}^k = y^k$. For the second factor on the right side of (27), we consider the cases $Y_{k+1} = 1$ and $Y_{k+1} = 0$ separately. In both cases, we will employ the binary random sequence W_k such that $W_k = 1$ if there is a packet that can be served at time k and otherwise $W_k = 0$. We can write $W_{k+1} = g_{k+1}(Y^k, X^{k+1})$, a deterministic function of Y^k and X^{k+1}. This implies

$$
\begin{aligned}
&P_{Y_{k+1}|Y^k, X^{k+1}}(y_{k+1}|y^k, x^{k+1}) \\
&= P_{Y_{k+1}|Y^k, X^{k+1}, W_{k+1}}(y_{k+1}|y^k, x^{k+1}, g_{k+1}(y^k, x^{k+1})). \quad (29)
\end{aligned}
$$

Moreover, there is a Markov chain

$$(Y^k, X^{k+1}) \to W_{k+1} \to Y_{k+1} \quad (30)$$

since

$$
\begin{aligned}
&P_{Y_{k+1}|Y^k, X^{k+1}, W_{k+1}}(y_{k+1}|y^k, x^{k+1}, 0) \\
&= P_{Y_{k+1}|W_{k+1}}(y_{k+1}|0) = \begin{cases} 1 & y_{k+1} = 0, \\ 0 & y_{k+1} = 1, \end{cases} \quad (31a) \\
&P_{Y_{k+1}|Y^k, X^{k+1}, W_{k+1}}(y_{k+1}|y^k, x^{k+1}, 1) \\
&= P_{Y_{k+1}|W_{k+1}}(y_{k+1}|1) = \begin{cases} 1 - p & y_{k+1} = 0, \\ p & y_{k+1} = 1. \end{cases} \quad (31b)
\end{aligned}
$$

It follows from (29) and (30) that

$$
\begin{aligned}
&P_{Y_{k+1}|Y^k, X^{k+1}}(y^{k+1}|y^k, x^{k+1}) \\
&= P_{Y_{k+1}|W_{k+1}}(y^{k+1}|g_{k+1}(y^k, x^{k+1})). \quad (32)
\end{aligned}
$$

Now consider the maximization of the second factor in (27). For the case $Y_{k+1} = 0$, it follows

$$
\begin{aligned}
&\max_{x^{k+1}} P_{Y_{k+1}|Y^k, X^{k+1}}(0|y^k, x^{k+1}) \\
&= \max_{x^{k+1}} P_{Y_{k+1}|W_{k+1}}(0|g_{k+1}(y^k, x^{k+1})) \leq 1, \quad (33)
\end{aligned}
$$

where the unity upper bound holds trivially. However, it follows from (31) that this unity upper bound is achieved by any x^{k+1} that ensures $W_{k+1} = 0$. Given $Y^k = y^k$, we observe that the upper bound in (33) is achieved by $(x^k, x_{k+1}) = (y^k, 0)$. In particular, given the output sequence y^k, the input $x^k = y^k$ implies that the system is idle at the end of slot k and, in this case, it follows that $x_{k+1} = 0$ implies $W_{k+1} = 0$. Thus for the case that $Y_{k+1} = 0$, we have shown that $P_{Y_{k+1}|Y^k, X^{k+1}}(\cdot|y^k, x^{k+1})$ is maximized over $x^{k+1} \in \mathcal{X}^{k+1}$ by $x^{k+1} = (x^k, x_{k+1}) = (y^k, 0) = y^{k+1}$.

Now for the case $Y_{k+1} = 1$, it follows from (31)

$$
\begin{aligned}
&\max_{x^{k+1}} P_{Y_{k+1}|Y^k, X^{k+1}}(1|y^k, x^{k+1}) \\
&= \max_{x^{k+1}} P_{Y_{k+1}|W_{k+1}}(1|g_{k+1}(y^k, x^{k+1})) \leq p. \quad (34)
\end{aligned}
$$

The upper bound is achieved by any x^{k+1} that ensures $W_{k+1} = 1$. No matter what the past history, $x^{k+1} = 1$ ensures $W_{k+1} = 1$. It follows that the upper bound in (34) is achieved by $(x^k, x_{k+1}) = (y^k, 1)$. Thus for the case that $Y_{k+1} = 1$, we have shown that $P_{Y_{k+1}|Y^k, X^{k+1}}(1|y^k, x^{k+1})$ is maximized over $x^{k+1} \in \mathcal{X}^{k+1}$ by $x^{k+1} = (x^k, x_{k+1}) = (y^k, 1) = y^{k+1}$.

Now returning to (27), the first factor is maximized by $\hat{x}^k = y^k$ and the second factor is maximized by $x^{k+1} = y^{k+1}$ hence the product is maximized by $x^{k+1} = (\hat{x}^k, x_{k+1}) = (y^k, 1) = y^{k+1}$.

Hence,

$$\max_{x^{k+1} \in X^{k+1}} P_{Y_{k+1}|Y^k, X^{k+1}}(y^{k+1}|y^k, x^{k+1}) = \mu^{y_{k+1}} \quad (35)$$

From (27), (28) and (35) we get,

$$\max_{x^{k+1} \in X^{k+1}} P_{Y^{k+1}|X^{k+1}}(y^{k+1}|x^{k+1}) = \mu^{\sum_{i=1}^{k+1} y_i}, \quad (36)$$

and the maximum is achieved when $x^{k+1} = y^{k+1}$.

Thus, the claim holds for $n = k + 1$, completing the proof of Lemma 1.