A Fresh Look at the Generalised Mix Framework

Andrei Serjantov[1]

The Free Haven Project (schnur@gmail.com)

Abstract. Anonymity systems designed to handle anonymous email have been implemented with a variety of different mixes. Although many of their properties have been analysed in previous work, some are still not well understood and many results are still missing.
In this paper we reexamine the generalised mix framework and the binomial mix of [7]. We show that under some parameterizations the binomial mix has undesirable properties. More specifically, for any constant parameterization of the binomial mix, there is a minimum number of messages beyond which it acts as a timed mix. In this case the number of messages inside it is no longer hidden from the adversary and the mix is vulnerable to easy active attack. We suggest ways to avoid this in the generalised mix framework. Secondly, we show that the binomial distribution used in the framework produces distribution of pool sizes with low variance and show how to improve on this.
Finally, we present a technique from queueing theory which allows us to analyse this property for a class of mixes assuming Poisson message arrivals.

1 Introduction

Anonymous email systems are commonly implemented using mixes [2]. To provide anonymity a mix has to follow a cryptographic protocol which ensures bitwise unlinkability to prevent attackers linking messages based on their bit patterns and a batching or reordering strategy to prevent timing attacks, i.e. adversaries linking messages by simply watching them coming in and out of the mix.

In this paper we consider batching strategies of mixes used in real message-based anonymity systems such as Mixmaster and Mixminion. In the remailer community which runs these systems there is an ongoing debate about the properties of different batching strategies; we hope this work not only contributes to this debate, but also helps influence the design of deployed systems and hence improve the anonymity properties for their users. We start off by describing what is perhaps the most sophisticated mix to date, the binomial mix.

The binomial mix has been proposed in [7] and further analysed in [4]. The batching strategy of this mix is as follows: if the mix contains M messages, then the number of messages to be forwarded on to their next hops (or destinations) is determined by the number of heads obtained from tossing a biased coin for each message. The bias of the coin is obtained from the function $g(M)$ which is the cumulative normal distribution function.

The rest of the paper is organized as follows: first we review the generalised mix framework. Then we look at the expected number of messages to be kept in the pool as a function of the number of messages in the mix for some existing mixes. We find that as the number of messages in the existing binomial mix increases, the expected size of the pool approaches zero and argue that this is undesirable. Another consequence of this is that the binomial mix loses its desirable property of hiding the number of messages inside it at high traffic volume. We then show that by altering the $g(M)$ function and the distribution from which the number of messages to be forwarded is drawn we can alter the expected size of the pool mix and its variance and hence retain the desirable properties of the binomial mix at high traffic volumes. Finally, we turn our attention to the distribution of the number of messages in the mix. We present a technique which allows us to calculate the distribution of messages inside the Stop and Go mix and to slight variants of the timed dynamic pool mix and the binomial mix assuming message arrivals are Poisson distributed.

2 The Generalised Mix Framework

The generalised mix framework and the binomial mix have evolved from the pool and the timed dynamic pool mixes [12]. The framework introduced two innovations: unlike in the case of the pool mixes where the number of messages to be forwarded is deterministic, it is now a random variable chosen from the binomial distribution[1], $\mathrm{Bin}(M, g(M))$. The expectation of this random variable is determined by a function g of the number of messages in the mix. Before we proceed, let us set up the terminology explicitly.

- M is the number of messages in the mix at the start of the round
- X is the number of messages retained in the mix
- $P = g(M)$ where $g : [0, \infty) \to [0, 1]$ is the probability of forwarding each message

Hence a mix is specified almost entirely[2] by the function $g(M)$. Whilst this is enough to express the mix strategy in a concise manner, we argue that it is more insightful to look at $P(X = x|M)$, the conditional distribution of the number of messages retained in the pool and its expectation and variance. Clearly, the number of messages which stay in the mix follows a binomial distribution $\mathrm{Bin}(M, 1 - g(M))$.

$$P(X = x|M) = \binom{M}{x} g(M)^{(1-x)}(1 - g(M))^x$$

$$\mathbb{E}[P(X = x|M)] = M(1 - g(M))$$

[1] Hence in most cases the attacker cannot tell with certainty how many messages are in the mix [7], but note [10].

[2] The only remaining parameter is t, how often the mix flushes.

$$\mathrm{Var}[P(X = x|M)] = Mg(M)(1 - g(M))$$

We now proceed to look closely at the expectation of the size of the pool in various existing mixes[3] defined in the generalised mix framework, i.e. via $g(M)$ and compare their properties.

3 Expected Pool Size of Various Mixes

We start by comparing the relatively simple mixes from [12] which were further analysed in [10, 11].

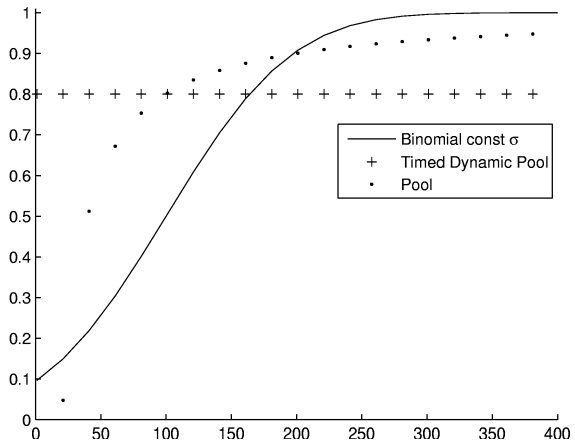

Fig. 1. The function $g(M)$ for some existing mixes

3.1 Timed Pool Mix

This mix always keeps n messages and outputs $M - n$. Note that although it is impossible to express exactly this behaviour in the binomial mix framework, it will suffice that the expected number of retained messages is n. See Figures 1 and 2 for a graphical representation of the properties of existing mixes.

$$g_p(M) = \begin{cases} 0 \text{ if } M \leq n \\ \frac{M-n}{M} \text{ otherwise} \end{cases}$$

[3] more precisely, their randomized versions

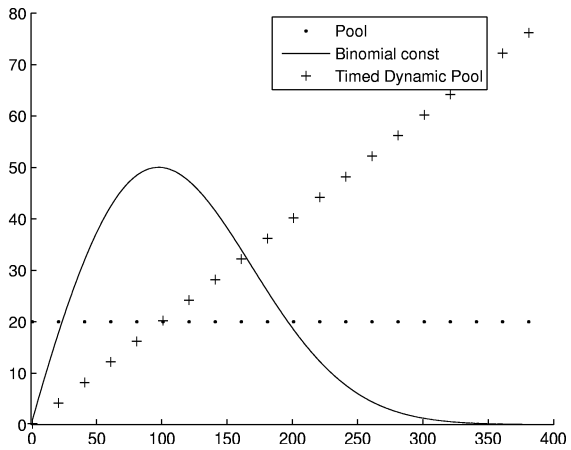

Fig. 2. Size of the pool as a function of messages in the mix

$$P(X = x|M) = \begin{cases} 0 \text{ if } M \leq n \\ M - n \text{ otherwise} \end{cases}$$

3.2 Timed Dynamic Pool Mix

This mix always keeps fM (where $f < 1$) messages, hence $g_{dp}(M) = f$ and outputs $(1 - f)M$. Sometimes a certain minimum is also specified, but this should be designed to act only very rarely in exceptional circumstance without changing the overall behaviour. Again, we have to make do with this being the expected number of messages output in our generalised framework. This clearly shows that the pool grows linearly with the number of messages in the mix. See Figures 1 and 2.

Clearly, the existing pool mixes define the limiting cases – a constant and a linear function. Let us now look at the binomial mix and see how it can be parameterized to behave as either of these.

4 The Binomial Mix

As mentioned above, the weight of the biased coin in the case of the binomial mix is determined from a cumulative distribution function of the normal distribution. The question that has not been addressed in the literature so far is *which* normal distribution. A normal distribution is uniquely defined by its mean and variance, $N(\mu, \sigma)$ hence the $g(M)$ of the binomial mix is as follows.

$$g(M, \mu, \sigma) = \int_{-\infty}^{M} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(M-\mu)^2}{2\sigma^2}}$$

Up to now it has been implicitly assumed that in $g(M, \mu, \sigma)$ μ and σ are independent of M (simply constants).

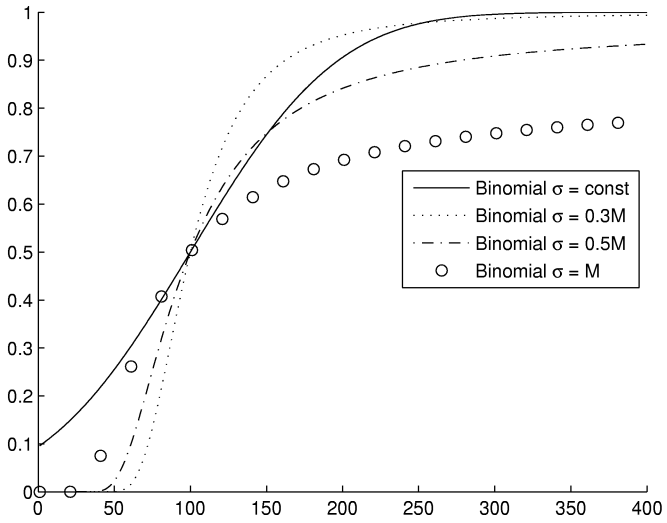

Fig. 3. Different parameterizations of the cumulative normal distribution function

The function does not have a closed form representation; we illustrate functions with $\sigma = 100, \sigma = 0.3M, \sigma = 0.5M, \sigma = M$. The difference between these may not seem significant, however, it is clearer when we examine the expected pool size as a function of M. This is illustrated in Figure 4.

Clearly, a binomial mix with a constant σ parameter $\lim_{M\to\infty} \mathbb{E}[P(X|M)] = 0$ approaches zero very quickly – the normal distribution has very thin tails. Hence at large M the constant-σ mix has turned into a simple timed mix. This is clearly undesirable: such a mix can be flushed in one round given a sufficiently high number of messages and hence admits an easy active attack [12].

The binomial parameterizations where σ is linear in M are much better. First, they retain the property of having a quickly increasing pool for small values of M, this can be adjusted via the μ parameter of the cumulative normal distribution function and behave like the timed dynamic mix at large values of M – linear pool size growth.

It is interesting to note that there are several alternatives, also expressed in the binomial mix framework. Hence below we present 3 mixes with 3 dif-

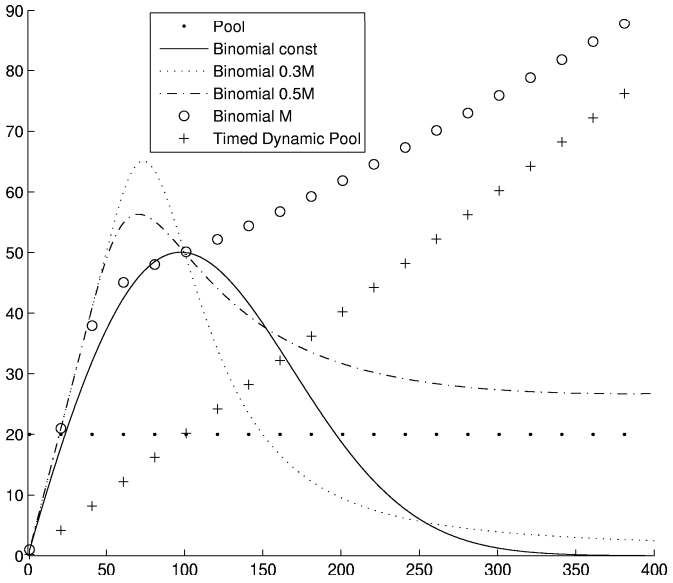

Fig. 4. Expected pool size for existing mixes and various parameterizations of the binomial mix

ferent properties: $\mathbb{E}[P(X|M)]$ approaching a non-zero constant (though this hardly helps with the active attack, the $g(M)$ function is simple and analytically tractable); logarithmic or square root growth. We show that in terms of the generalised mix definition they look quite similar, hence looking at the growth of the size of the pool has been an insightful exercise.

Properties of the new mixes are shown in Figure 5. Because the mixes are all expressed in the generalised framework, their anonymity and delay properties (although not in closed form) follow directly from [7, 4, 5]; we do not restate them here.

4.1 Binomial+ Mix

First we try to find a mix which is similar to the binomial mix, but can be adapted so that $\lim_{M\to\infty} P(M) = n$, i.e. it has a pool of at least n messages. We find the following function to be suitable:

$$g(M) = 1 - \frac{(M-n)e^{-kM} + n}{M}$$

$$\mathbb{E}[P(X|M)] = (M-n)e^{-kM} + n$$

Figure 5 uses $k = 0.01$. Indeed, $\mathbb{E}[P(X|M)]$ has a similar shape to that of the binomial mix.

4.2 Logarithmic and Square Root Mixes

If we seek to have slow growth of the pool size, we can have a logarithmic or a square root function for $\mathbb{E}[P(X|M)]$ and hence have $g_{\log}(M) = 1 - \frac{\log(M)}{M}$ and $g_{\text{sqrt}}(M)) = 1 - \frac{1}{\sqrt(M)}$. Their behaviour is shown in Figure 5. A practical implementation may have lower bound on the size of the pool in either case; here we are concerned with the asymptotic properties only. Clearly, these mixes have higher expected delay but also higher anonymity.

5 Distribution of the Number of Messages to Forward

One of the benefits of introducing the binomial framework as presented in [7] is the fact that the binomial mix hides the number of messages inside it which makes it (slightly) more difficult to mount a blending attack on it[4]. And yet the obvious parameterization implies that (as the authors of the original paper conjecture) by sending a sufficiently high number of messages during a single round, all messages can be flushed from the mix with a high probability. Making this more precise, the probability of having a message retained in the mix is:

$$P(X \geq 1|M) = 1 - (g(M))^M$$

[4] For a thorough analysis of similar issues for the existing mixes see [10]

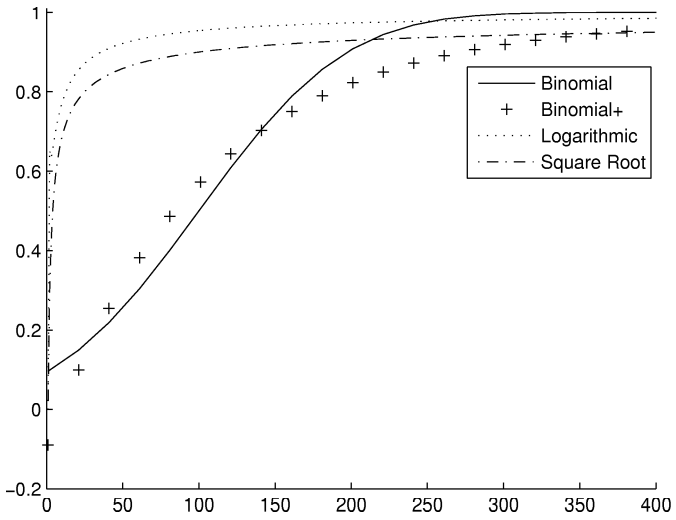

Fig. 5. The function $g(M)$ for some new mixes

It is evident that this probability approaches 1 for the Logarithmic, the Square root and the Binomial+ mixes, and hence it is impossible to flush the mix in a single round and mount an easy active attack. What is less obvious (due to the lack of closed form representation of $g(M)$) is that this probability asymptotically approaches 0 for the binomial mix with constant σ[5]. Clearly, as we have seen above, this is not the case when σ is a function of M.

Attacking the same problem from a slightly different angle, if we examine the variance of the distribution $P(X|M)$, we find that while the variance for the number of messages sent out by the constant-σ binomial mix approaches zero with increasing M, it increases in the case of the other mixes. The expectation $\mathbb{E}[P(X|M)]$ and values of one standard deviation around it for the Binomial with constant σ, Binomial with $\sigma = 0.5M$ are shown in Figure 6.

It is also clear that the variance of the pool size in the other mixes does not grow significantly as M increases. This naturally suggests a rethink of the generalised mix framework; instead of specifying $g(M)$, the mix should be defined by the probability distribution of the number of messages to be forwarded given the number of messages in the pool, $P(X = x|M)$, with $x \in \{0 \dots M\}$[6]. We have already seen the case where $P(X = x|M) = \text{Bin}(M, 1 - g(M))$, however there are many alternatives: the Hypergeometric distribution, the Uniform distribution, the Maximum Entropy Distribution which we describe further below or (the inelegant) discretized versions of the scaled Beta or the Normal distribu-

[5] The reader is invited to verify this either by analytical or empirical means

[6] Given X, we construct a random permutation of messages and forward the first X

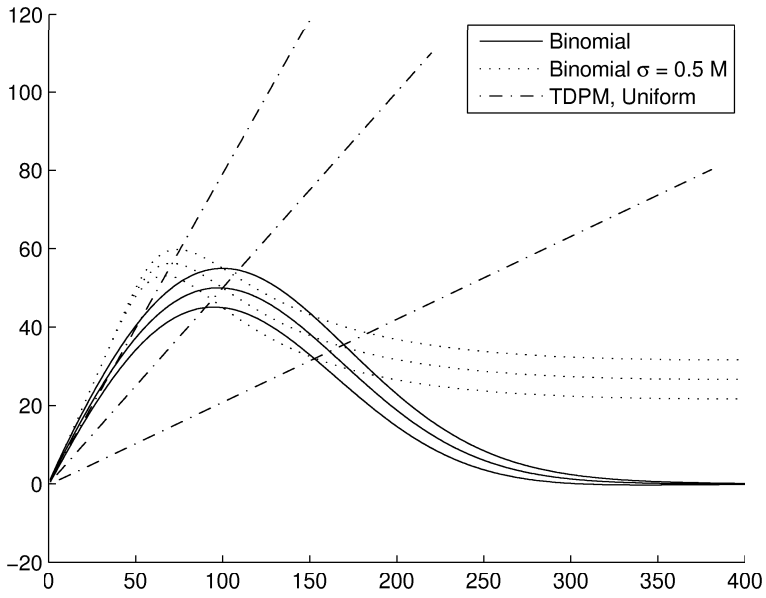

Fig. 6. Comparing Expectations and Standard Deviations of Pool Size

tions. Note, however, that from above we already have good candidates for the expectation of such distributions, namely the linear σ binomial, square root or logarithmic mix $g(M)$ functions.

We do not delve into the question of distributions too deeply, but show by example that the difference in the variance of the pool size is substantial and present the maximum entropy distribution which maximizes variance. We conjecture that such a distribution is optimal at hiding the number of messages in the mix.

Example 1. In this paper we described a version of the timed dynamic pool mix which had

$$\lfloor P(X|M) \rfloor = 0.5M$$
$$P(X = x|M) = \binom{M}{x} f^{(1-x)}(1-f)^x$$

Instead, we could have a timed dynamic pool mix with the same $\lfloor P(X|M) \rfloor = 0.5M$ but $P(X = x|M) = \text{Uniform}[0, M]$. The expectation value and the values one standard deviation away from it of such a timed dynamic pool mix are also shown on Figure 6.

The variance of the uniform distribution is $\frac{(M+1)^2-1}{12}$ which is greater than that of the binomial distribution, $M/4$. Note that given a set of values $\{0 \dots M\}$, and a given expected value μ the maximum entropy distribution is of the following form:

$$P(X = x) = Cr^x$$

Using the facts that the sum of the probabilities equals 1 and the expectation equals μ allows us to determine the values for constants C and r. For example, if we have 100 messages in the mix, we may flush between zero and 100. We wish to use the maximum entropy distribution to determine how many should be flushed, with the expected number set at 20 messages. Using numerical methods to obtain C and r, we find that the distribution to use is as follows:

$$P(X = x) = 1.58342(-0.986546)^x$$

Naturally, if $\mu = M/2$, the maximum entropy distribution is simply the uniform distribution.

6 Distribution of the Number of Messages in Mixes

The number of messages inside simple mixes during operation is well understood. For example, the threshold mix contains no more than N messages, the timed mix contains quite simply all the messages which have arrived since the last flush, the timed pool mix contains all the messages which have arrived since the last flush plus n, the size of the pool. For more complex mixes, this number or

rather, the distribution of the number of messages inside the mix is not so clear. Yet a mix can only store a finite number of messages, so this distribution needs to be understood in order to minimize the probability of a message having to be dropped. This, in part, has originally motivated the choice of $g(M)$ of the binomial mix which makes it behave as a simple timed mix at high loads.

In the first part of this paper we showed that both the timed dynamic pool mix and the improved parameterization of the binomial mix retain a constant fraction of messages – they both have the property that $lim_{M\to\infty} g(M) = c$ for $c < 1$. In this section we present a method for determining the distribution of the number of messages inside various mixes, in particular the Timed Dynamic Pool Mix. Such a method allows us to to determine the probability of the mix running out of space and hence select a suitable parameterization to avoid this.

First, we consider Stop and Go Mix first introduced by Kesdogan in [9]. It delays each message individually by an amount picked from an exponential distribution. Assuming Poisson distribution of message arrivals, we can model it as an $M/M/n$ process and use standard queuing theory techniques [3] as we informally outline below.

We proceed by denoting a mix as an n-state system where $n-1$ is the maximum possible number of messages in the mix. We assume message arrivals are distributed with a Poisson distribution with parameter λ and the time between flushes is distributed exponentially with parameter μ. The system changes state when either one message arrives (with probability λ) or one message leaves (with probability μ). For example, take a mix which can hold a maximum of three messages. The rates of transitions between states 0 to 3 (0,1,2 or 3 messages inside the mix) are as follows:

$$A_{sg} = \begin{pmatrix} -\lambda & \mu & 0 & 0 \\ \lambda & -(\lambda+\mu) & \mu & 0 \\ 0 & \lambda & -(\mu+\lambda) & \mu \\ 0 & 0 & \lambda & -\mu \end{pmatrix}$$

The rows represents the state of the mix. We see that the rate of transition out of state 0 and into to state 1 is λ – this is the probability that a message arrives at the mix. Similarly, the rate of transition into state 0 from state 1 is μ. Reading row 2, the rate of transition into state 1 is λ from state 0, μ from state 3 and $-(\lambda+\mu)$ to account for the probabilities of a message arriving or leaving while the mix is in state 2.

Now, we seek a vector of probabilities P such that the system is in equilibrium, i.e. there is no net inflow or outflow from each state. Our last constraint is that the probabilities sum to 1. We now solve the system of linear equations $AP = 0$ together with $\sum P = 1$ and obtain P, the probabilities of finding the system in each state. For instance, when $\lambda = 1/3$ and $\mu = 1/2$, the distribution of messages in the 4 state mix defined above is: $[0.4154, 0.2769, 0.1846, 0.1231]$.

The same technique can be used directly for a mix with exponential inter-flush times but which, like the timed dynamic pool mix, flushes deterministic batches of messages. The transition matrix for this mix ($f = 0.5$) differs from

the one above only by the position of the μ in the right hand column – the mix transitions from having 3 messages inside it to having 1.

$$A_{tdpm} = \begin{pmatrix} -\lambda & \mu & 0 & 0 \\ \lambda & -(\lambda+\mu) & \mu & \mu \\ 0 & \lambda & -(\mu+\lambda) & 0 \\ 0 & 0 & \lambda & -\mu \end{pmatrix}$$

The vector of probabilities is now $[0.4737, 0.3158, 0.1263, 0.0842]$. The probabilities of high states are lower because more messages get forwarded on some of the flushes, hence fewer remains in the mix. As a further example of the capabilities of this technique, we calculated the distribution of the number of messages inside a pool and a timed dynamic pool mixes, each with maximum capacity 100 messages. These are illustrated in Figure 7.

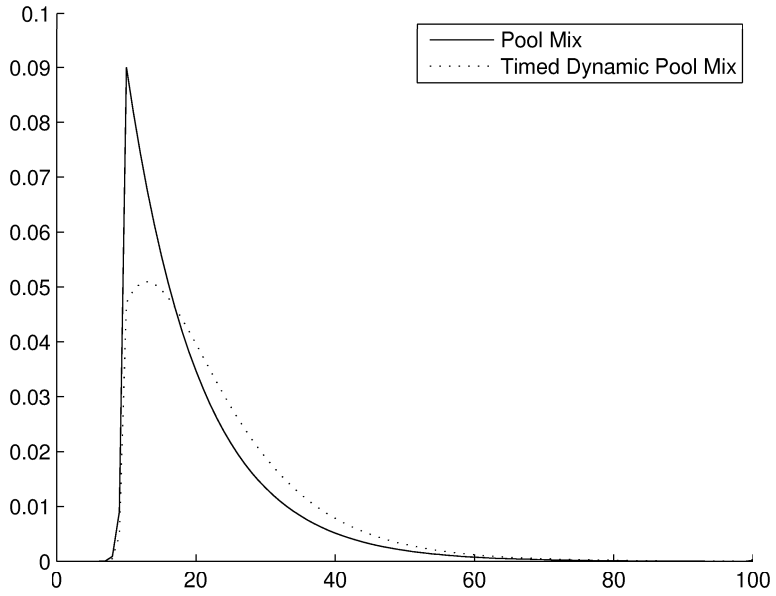

Fig. 7. Distributions of the number of messages inside mixes. $\lambda = 5$, $\mu = 0.5$

This technique is efficient and allows us to calculate the probability distribution of the number of messages inside mixes with arbitrary $P(X = x|M)$ with exponential inter-flush times in an environment with Poisson-distributed inter-arrival times. Hence a slight modification of the binomial mix with exponential inter-flush times falls into this category and can now be analysed.

More advanced queueing theory tools are needed to consider other known mixes. To be more precise, mixes with Poisson arrivals and arbitrary distributions of inter-flush times can be described by a $M/G/n$ model and those with deterministic inter-flush times (for instance Pool or Timed Dynamic Pool mixes) fit the $M/D/s$ model. The interested reader is invited to refer to [3].

In this section we assumed that message inter-arrival times are Poisson distributed. To the best of our knowledge, the only work investigating the issue is [6]; there the authors find a large structural break in their data sample. We briefly reexamined the same data and looking at shorter time horizons we find the distribution broadly Poisson; though a full empirical investigation of inter-arrival times is long overdue we do not consider the issue here. From the theoretical point of view, the number of messages inside a mix which forwards a constant fraction of messages (such as the linear-σ binomial or the timed dynamic pool mix) follows a mean reverting Ornstein-Uhlenberg stochastic process with non-Gaussian increments (the increments model the distribution of the number of messages arriving in one batch). Theories of such processes with arbitrary increments exist [7]; in particular it is reassuring that under reasonable assumptions the implied distributions inside the mixes are stationary. The mathematically inclined reader is referred to [1, 8] for (very complex) properties of such processes.

7 Conclusion

In this paper we drew attention to the asymptotic properties of mixes. By considering how the size of the pool mix grows with the number of messages in the mix, we showed that the obvious previously used parameterization of the binomial mix has some undesirable properties and proposed a fix. We have also suggested some new mixes within the generalised mix framework. Next, we showed that the variance of the previously used binomial mix is zero at high loads, hence it no longer has pool size hiding properties. Furthermore, mixes which use the generalised mix framework all have small variance of pool size. We propose using arbitrary distributions for pool size and show how this can increase the variance of the pool size. Finally, we present a method for determining the distribution of messages inside various mixes assuming Poisson message arrivals.

References

1. O.E. Barndorff-Nielsen and N. Shepard. Non-gaussian OU based models and some of their uses in financial economics and modelling by Levy processes for financial econometrics. Economics Papers 1999-w9/2000-w3, Economics Group, Nuffield College, University of Oxford, 2000.
2. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.
3. R. Cooper. *Introduction to Queueing Theory.* New York: North-Holland, 1981.

[7] they turn out to be useful in modeling stochastic volatility and electricity prices(!)

4. Claudia Díaz. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, December 2005.
5. Claudia Díaz and Bart Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In *Proceedings of 6th Information Hiding Workshop (IH 2004)*, LNCS, Toronto, May 2004.
6. Claudia Díaz, Len Sassaman, and Evelyne Dewitte. Comparison between two practical mix designs. In *Proceedings of ESORICS 2004*, LNCS, France, September 2004.
7. Claudia Díaz and Andrei Serjantov. Generalising mixes. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760, March 2003.
8. Lancelot F. James. Laws and likelihoods for Ornstein Uhlenbeck-Gamma and other BNS OU stochastic volatilty models with extensions, 2006. ⟨`http://www.citebase.org/abstract?id=oai:arXiv.org:math/0604086`⟩.
9. Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525, 1998.
10. Luke O'Connor. On blending attacks for mixes with memory. In *Proceedings of Information Hiding Workshop (IH 2005)*, June 2005.
11. Andrei Serjantov. *On the Anonymity of Anonymity Systems*. PhD thesis, University of Cambridge, June 2004.
12. Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Proceedings of Information Hiding Workshop (IH 2002)*. Springer-Verlag, LNCS 2578, October 2002.