Proofs of the Propositions

The proofs of the propositions stated in the paper titled "LeakyRand: An Efficient Covert Channel in Fully Associative Last-level Caches with Random Eviction" are discussed below.

Proposition#1: A fully-associative cache of capacity c blocks exercises random replacement with a uniform victimization probability of $\frac{1}{c}$ for any cache block. Any access sequence having n cache misses achieves an expected occupancy of $c - c\left(1-\frac{1}{c}\right)^n$ blocks in such a cache with variance in the achieved occupancy being $c\left(1-\frac{1}{c}\right)^n + c(c-1)(1-\frac{2}{c})^n - c^2\left(1-\frac{1}{c}\right)^{2n}$.

Proof: Let us number the blocks in the cache from 0 to $c-1$. Let $X_i = 1$ if cache block i is occupied by some block from the occupancy sequence at the end; otherwise X_i is 0. Define $X = \sum_{i=0}^{c-1} X_i$, which is equal to the total number of cache blocks occupied by the occupancy sequence at the end. So, $E[X] = E[\sum_{i=0}^{c-1} X_i] = \sum_{i=0}^{c-1} E[X_i]$ by linearity of expectation. Now, $E[X_i] = P(X_i = 1) = 1 - P(X_i = 0)$. We observe that X_i is 0 if cache block i remains unoccupied by the occupancy sequence. This is possible only if all n misses replace blocks other than cache block i. Therefore, $P(X_i = 0) = \left(\frac{c-1}{c}\right)^n$ assuming the n replacement events to be independent which is usually how a random replacement policy is expected to be implemented. So, $E[X] = \sum_{i=0}^{c-1}\left\{1 - \left(\frac{c-1}{c}\right)^n\right\} = c - c\left(1-\frac{1}{c}\right)^n$.

The variance of X is given by $E[X^2] - (E[X])^2$. Now, $X^2 = \sum_{i=0}^{c-1} X_i^2 + 2\sum_{i<j} X_i X_j$. Therefore, by linearity of expectation, $E[X^2] = \sum_{i=0}^{c-1} E[X_i^2] + 2\sum_{i<j} E[X_i X_j]$. Since X_i takes values 0 and 1 only, $E[X_i^2] = E[X_i]$ and hence, $\sum_{i=0}^{c-1} E[X_i^2] = \sum_{i=0}^{c-1} E[X_i] = E[X]$. We note that $E[X_i X_j] = P(X_i = X_j = 1) = P(X_i = 1) - P(X_i = 1, X_j = 0) = 1 - P(X_i = 0) - (P(X_j = 0) - P(X_i = 0 = X_j)) = 1 - 2\left(1-\frac{1}{c}\right)^n + \left(1-\frac{2}{c}\right)^n$. Therefore, $2\sum_{i<j} E[X_i X_j] = 2\binom{c}{2}\left(1 - 2\left(1-\frac{1}{c}\right)^n + \left(1-\frac{2}{c}\right)^n\right)$.

$Variance[X] = E[X^2] - (E[X])^2 = E[X] + 2\binom{c}{2}\left(1 - 2\left(1-\frac{1}{c}\right)^n + \left(1-\frac{2}{c}\right)^n\right) - (E[X])^2$. Plugging in the expression for $E[X]$ and simplifying, we get $Variance[X] = c\left(1-\frac{1}{c}\right)^n + c(c-1)(1-\frac{2}{c})^n - c^2\left(1-\frac{1}{c}\right)^{2n}$. □

Observation: For a cache with 2 MB capacity and 64-byte blocks, we have $c = 32768$. For this value of c, we numerically find that the variance attains the maximum value of about 3336 for $n = 41170$. Thus, the standard deviation in occupancy for a 2 MB cache remains bounded by 57.76 cache blocks which is only 0.18% of the cache capacity.

Proposition#2: Let the LLC capacity be c blocks and the Occupancy Set size be $|OS|$. If k is the number of blocks other than the Occupancy Set blocks accessed in the `while` loop starting at the label `ProbeAndFlush` of Algorithm **??**, the probability that the value of the variable `occupancy` is exactly equal to the number of invalid LLC ways created during this loop is at least $\left(1-\frac{k}{c}\right)\left(1-\frac{5k}{c}\right)\left(1 - \frac{k}{c-k}\left(1-\left(\frac{k}{c}\right)^{|OS|-2}\right)\left(\frac{3k}{c}+2\right)\right)$.

Proof: In an iteration of the `while` loop, the value of the `occupancy` variable may remain the same (event X) or may increase by one (event Y), while the number of invalid LLC ways may remain the same (event A), may increase by one (event B), or may decrease by one or more (event C). Therefore, there are six events to consider in an iteration of the `while` loop: (XA), (XB), (XC), (YA), (YB), (YC). These are mutually exclusive and exhaustive. Among these, the events (XA) and (YB) keep the value of the `occupancy` variable and the number of invalid LLC ways equal. Therefore, we need to bound the probability of the remaining four events. The `while` loop runs for $|OS|$ iterations. Let us number the iterations $1, 2, \ldots, |OS|$. Thus, the n^{th} iteration corresponds to `i=` $|OS| - n$. We note that event (X) cannot take place in the first iteration because the `occupancy` variable is always incremented in that iteration. Also, event C cannot take place in the first iteration because we assume that initially the LLC has no invalid ways and hence, the number of invalid LLC ways cannot decrease in the very first iteration. Therefore, the events (XA), (XB), (XC), (YC) cannot take place in the first iteration.

Event (XB) may take place during the n^{th} iteration ($n > 1$) in two situations: (a) `load(OccupancySet[i])` suffers an LLC miss AND there is no invalid LLC way at the time of filling `OccupancySet[i]` in LLC AND `OccupancySet[i]` does not get replaced from LLC between the `load` and the `clflush`, (b) `load(OccupancySet[i])` experiences an

LLC hit which is inferred as an LLC miss because the code block containing `rdtsc` suffers an LLC miss AND `OccupancySet[i]` does not get replaced from the LLLC between the `load` and the `clflush`. Cleary, the first situation is dependent on the absence of any invalid LLC way up to the n^{th} iteration. The second situation is also dependent on the same condition because for the code block containing `rdtsc` to get replaced by an LLC miss, there cannot be any invalid way in the LLC. Now, absence of an invalid LLC way in the n^{th} iteration implies that all iterations up to that point failed to create an invalid LLC way. This is possible only if `OccupancySet[j]` got replaced from the LLC between `load(OccupancySet[i])` and `clflush(OccupancySet[i])` for all iterations `i` up to that point and this replacement happens because one of the k blocks not belonging to the Occupancy Set suffers an LLC miss. Since the probability that an LLC miss replaces `OccupancySet[j]` is $\frac{1}{c}$, the probability that any of the k non-Occupancy Set blocks replaces `OccupancySet[i]` for all $n-1$ iterations is at most $\left(\frac{k}{c}\right)^{n-1}$. Thus, the probability of event (XB) in the n^{th} iteration is at most $2\left(\frac{k}{c}\right)^{n-1}$ for $n > 1$.

Event (XC) can happen in the n^{th} iteration if at least one non-Occupancy Set block B suffers a miss and fills up an existing invalid LLC way. This is possible only if B is replaced in the $(n-1)^{\text{th}}$ iteration before any invalid LLC way is created, an invalid LLC way then gets created in the $(n-1)^{\text{th}}$ iteration, and finally B is accessed in the n^{th} iteration which fills up the invalid LLC way. Therefore, the probability of event (XC) is at most the probability that no invalid LLC way is created up to $(n-2)^{\text{th}}$ iteration. This probability is at most $\left(\frac{k}{c}\right)^{n-2}$ for $n > 2$, as already discussed above. The event (XC) can happen also for iteration $n = 2$. In this case, an invalid LLC way is created in iteration $n = 1$ which is filled up in iteration $n = 2$ by the block B which was replaced in iteration $n = 1$. The probability of an LLC miss replacing B in iteration $n = 1$ is at most $\frac{k}{c}$ as B can be any of the k non-Occupancy Set blocks.

Event (YA) takes place in the first iteration ($n = 1$) if `OccupancySet[0]` is replaced from the LLC between `load(OccupancySet[0])` and `clflush(OccupancySet[0])`. It can also happen in the n^{th} iteration for $n > 1$ if `load(OccupancySet[i])` is a hit AND `OccupancySet[i]` gets replaced from the LLC between the `load` and the `clflush`. As already discussed, such a replacement would happen only due to the absence of an invalid LLC way in the cache in the n^{th} iteration. Thus, the probability of event (YA) is at most $\left(\frac{k}{c}\right)^{n-1}$ in iteration $n > 1$. The probability of event (YA) in iteration $n = 1$ is at most $\frac{k}{c}$ because any of the k blocks can replace `OccupancySet[0]` in that iteration.

Event (YC) takes place in the n^{th} iteration if `load(OccupancySet[i])` is a hit AND at least one non-Occupancy Set block suffers an LLC miss and fills up an existing hole. The calculation of the probability of event (YC) follows the same argument as that of event (XC). Thus, the probability of event (YC) is at most $\left(\frac{k}{c}\right)^{n-2}$ in iteration $n > 2$ and $\frac{k}{c}$ in iteration $n = 2$.

The probability that at least one of the events (XB), (XC), (YA), and (YC) takes place in iteration $n > 2$ is at most $3\left(\frac{k}{c}\right)^{n-1} + 2\left(\frac{k}{c}\right)^{n-2}$. The probability that at least one of these events takes place in iteration $n = 2$ is at most $5\frac{k}{c}$. The probability that at least one of these events takes place in iteration $n = 1$ is at most $\frac{k}{c}$ (only event (YA) can take place in iteration $n = 1$). Therefore, the probability that none of these events takes place in any of $|OS|$ iterations is at least $P = \left(1 - \frac{k}{c}\right)\left(1 - 5\frac{k}{c}\right)\Pi_{n=3}^{|OS|}\left(1 - 3\left(\frac{k}{c}\right)^{n-1} - 2\left(\frac{k}{c}\right)^{n-2}\right)$. Now, the difference between the value of the `occupancy` variable (say, V) and the number of invalid LLC ways (say, W) starts off at zero. It remains zero if none of the events (XB), (XC), (YA), and (YC) takes place in any of the iterations or they take place according to certain patterns so that $V - W$ remains zero (e.g., positive and negative differences in $V - W$ are equal). Thus, the probability that $V - W$ is zero is at least as large as the probability that none of the events (XB), (XC), (YA), and (YC) takes place in any of the iterations, which is at least P. Next, we will simplify the expression for P and obtain a slightly less tighter bound which is easier to evaluate, but sufficiently tight to serve our purpose.

It is easy to prove by induction on m that $\Pi_{i=1}^{m}(1 - x_i) \geq 1 - \Sigma_{i=1}^{m} x_i$ for $1 \geq x_i \geq 0$. Therefore, $P \geq \left(1 - \frac{k}{c}\right)\left(1 - \frac{5k}{c}\right)\left(1 - \Sigma_{n=3}^{|OS|}\left(3\left(\frac{k}{c}\right)^{n-1} + 2\left(\frac{k}{c}\right)^{n-2}\right)\right)$. Substituting $\Sigma_{n=3}^{|OS|}\left(\frac{k}{c}\right)^{n-1} = \left(\frac{k}{c}\right)^{2}\frac{c}{c-k}\left(1 - \left(\frac{k}{c}\right)^{|OS|-2}\right)$ and $\Sigma_{n=3}^{|OS|}\left(\frac{k}{c}\right)^{n-2} = \frac{k}{c}\frac{c}{c-k}\left(1 - \left(\frac{k}{c}\right)^{|OS|-2}\right)$, we get $P \geq \left(1 - \frac{k}{c}\right)\left(1 - \frac{5k}{c}\right)\left(1 - \frac{k}{c-k}\left(1 - \left(\frac{k}{c}\right)^{|OS|-2}\right)\left(\frac{3k}{c} + 2\right)\right)$. □

Proposition#3. The LeakyRand channel is used to communicate a binary string of length n where each bit position is equally likely to be 0 or 1. If the Occupancy blocks fill up a fraction f of the LLC, the expected 1 bit to 0 bit error rates for Disturbance Set sizes one and two are respectively $\frac{1}{2} - \frac{f}{n(1-f)}\left[1 - \left(\frac{1+f}{2}\right)^{n}\right]$ and $\frac{1}{2} - \frac{f(f+2)}{n(1-f^2)} + \frac{2f}{n(1-f)}\left(\frac{1+f}{2}\right)^{n} - \frac{f^2}{n(1-f^2)}\left(\frac{1+f^2}{2}\right)^{n}$.

Proof: For a Disturbance Set of size one, a 1 bit to 0 bit error occurs when the sender evicts a block that is not an

Occupancy block. Let us suppose that it happens when transmitting the bit at position k. This means that the value of the bit at position k must be 1. Since all subsequent 1 bits will be transmitted as 0 bits, there would be exactly $m+1$ errors if there are m 1 bits in the remaining $n-k$ bits. The probability that there are m 1 bits in $n-k$ bits is $\binom{n-k}{m}\frac{1}{2^{n-k}}$. If we refer to the probability of evicting a block that is not an Occupancy block when transmitting the k^{th} bit as $P(k)$, then the probability of exactly $m+1$ errors is $\Sigma_{k=1}^{n} P(k)\binom{n-k}{m}\frac{1}{2^{n-k}}$. So, the expected number of errors is $\Sigma_{m=0}^{n-1}(m+1)\Sigma_{k=1}^{n} P(k)\binom{n-k}{m}\frac{1}{2^{n-k}} = \Sigma_{k=1}^{n}\Sigma_{m=0}^{n-k}(m+1)P(k)\binom{n-k}{m}\frac{1}{2^{n-k}} = \Sigma_{k=1}^{n}\frac{P(k)}{2^{n-k}}\Sigma_{m=0}^{n-k}(m+1)\binom{n-k}{m}$. Differentiating the expansion of $x(1+x)^{n-k}$ with respect to x and putting $x=1$, we obtain $\frac{P(k)}{2^{n-k}}\Sigma_{m=0}^{n-k}(m+1)\binom{n-k}{m} = P(k)+\frac{1}{2}(n-k)P(k)$. So, the expected number of errors is $\Sigma_{k=1}^{n}(P(k)+\frac{1}{2}(n-k)P(k))$. The probability that an invalid LLC way gets created in the region of the LLC not occupied by the Occupancy blocks when transmitting a bit is equal to the product of the probabilities that the bit is 1 and that the sender evicts a block from the $(1-f)$ fraction of the LLC. So, this probability is $\frac{1-f}{2}$. Therefore, $P(k) = (1-\frac{1-f}{2})^{k-1}\frac{1-f}{2} = \left(\frac{1+f}{2}\right)^{k-1}\frac{1-f}{2}$. Thus, the expected number of errors simplifies to $\frac{n}{2}-\frac{f}{1-f}\left[1-\left(\frac{1+f}{2}\right)^{n}\right]$. Hence, the expected bit error rate is $\frac{1}{2}-\frac{f}{n(1-f)}\left[1-\left(\frac{1+f}{2}\right)^{n}\right]$.

For a Disturbance Set of size two, a 1 bit to 0 bit error occurs when the sender evicts two blocks that are not Occupancy blocks leading to the creation of two invalid LLC ways. This can happen when transmitting the k^{th} bit ($k \geq 1$) if (i) both invalid ways are created when transmitting this bit, or (ii) one invalid way is created when transmitting this bit and another is created when transmitting an earlier bit. The probability of the first of these two events is $\left(\frac{1}{2}+\frac{1}{2}f^2\right)^{k-1}\frac{(1-f)^2}{2}$. The second event can be constructed by creating the first invalid way when transmitting the m^{th} bit ($m < k$) and then creating the second invalid way when transmitting the k^{th} bit. The probability of this event is $\Sigma_{m=1}^{k-1}\left(\frac{1}{2}+\frac{1}{2}f^2\right)^{m-1}\frac{1}{2}2f(1-f)\left(\frac{1}{2}+\frac{1}{2}f\right)^{k-1-m}\frac{1-f}{2}$, which simplifies to $\frac{1-f}{2^{k-1}}[(1+f)^{k-1}-(1+f^2)^{k-1}]$. So, the probability that two invalid LLC ways are created outside the region occupied by the Occupancy blocks when transmitting the k^{th} bit is $P(k) = \frac{1-f}{2^{k-1}}[(1+f)^{k-1}-(1+f^2)^{k-1}]+\frac{(1-f)^2}{2^k}(1+f^2)^{k-1} = (1-f)\left(\frac{1+f}{2}\right)^{k-1} - \left(\frac{1-f^2}{2}\right)\left(\frac{1+f^2}{2}\right)^{k-1}$. After two invalid LLC ways are created, all subsequent 1 bits will be transmitted as 0 bits. So, there will be exactly $m+1$ errors if there are m 1 bits in the subsequent transmission of $n-k$ bits. The probability that there are m 1 bits in $n-k$ bits is $\binom{n-k}{m}\frac{1}{2^{n-k}}$. As we have already shown, the expected number of errors is $\Sigma_{k=1}^{n}(P(k)+\frac{1}{2}(n-k)P(k))$. Substituting $P(k)$ and simplifying, we obtain the expected number of errors $= \frac{n}{2}-\frac{f(f+2)}{1-f^2}+\frac{2f}{1-f}\left(\frac{1+f}{2}\right)^{n}-\frac{f^2}{1-f^2}\left(\frac{1+f^2}{2}\right)^{n}$. Hence, the expected bit error rate is $\frac{1}{2}-\frac{f(f+2)}{n(1-f^2)}+\frac{2f}{n(1-f)}\left(\frac{1+f}{2}\right)^{n}-\frac{f^2}{n(1-f^2)}\left(\frac{1+f^2}{2}\right)^{n}$. □