V. Theoretical Analysis

A. Ratio of Seesaw Counters with Non-empty Negative Cell

Lemma V.1. *The ratio of (non-empty negative cell) seesaw counters is $\mu \approx 1 - e^{-(\frac{k}{B})(\frac{N_n}{N_p})}$ in expectation, where k is the number of hash functions, B is the counter per cell in SCA, N_n is the number of negative keys to be encoded, and N_p is the capacity of positive keys in SSCF.*

Proof. Considering that there are N_n negative keys, the probability that a seesaw counter is not mapped by all negative keys is $1 - (1 - \frac{1}{m})^{k \cdot N_n}$. Therefore, the ratio of negative seesaw counters is $\mu = 1 - (1 - \frac{1}{m})^{k \cdot N_n} \approx 1 - e^{-\frac{kN_n}{m}}$ in expectation. Besides, considering that the counter per key is B in SCA, then $\mu \approx 1 - e^{-\frac{k \cdot N_n}{B \cdot N_p}} = 1 - e^{-(\frac{k}{B})(\frac{N_n}{N_p})}$. □

B. Hash Modulating

Lemma V.2. *After inserting N_p positive keys, the expected number of triggered hash modulating is $\varphi = N_p \cdot (1 - (1 - \mu)^k)$.*

Proof. With one positive key to be inserted, the hash function modulating is triggered if one or more applied hash functions map the inserted key to a negative seesaw counter, whose probability is $1 - (1 - \mu)^k$. By the linearity of expectation, with N_p positive keys, the expected number of triggered hash modulating is $\varphi = N_p \cdot (1 - (1 - \mu)^k)$. □

Lemma V.3. *Suppose there are φ times of hash modulating and $n = s \cdot \varphi$ cells in HashModulator, the load factor of HashModulator is $\mathcal{L} = (1 - \mathrm{e}^{-\frac{1}{s}})$, where s is a constant scale factor that varies the space size of HashModulator.*

Proof. Considering that each key is mapped to HashModulator with a public hash function (*i.e.*, h_0), the load factor of HashModulator is then equivalent to the number of cells being occupied. We start with considering probability of the complementary case, *i.e.*, a cell is not mapped by all keys to be modulated, which is given by $(1 - \frac{1}{s \cdot \varphi})^{\varphi}$. That is to say, a cell will be occupied with probability $1 - (1 - \frac{1}{s \cdot \varphi})^{\varphi}$. By the linearity of expectation, the load factor $\mathcal{L}$ is $(1 - (1 - \frac{1}{s \cdot \varphi})^{\varphi}) \approx (1 - \mathrm{e}^{-\frac{\varphi}{s \cdot \varphi}}) = (1 - \mathrm{e}^{-\frac{1}{s}})$. This completes the proof. □

Lemma V.4. *When the load factor of HashModulator is $\mathcal{L}$, the probability that one hash function modulating is successful is $P_{ms} = \mathcal{L}(1 - \mu) + (1 - \mathcal{L})(1 - \mu^{2^{\eta_2}})$, where μ is the ratio of negative seesaw counters in SCA, η_2 is ModulatedIndex field size in HashModulator.*

Proof. The hash modulating is successful if it switches one initial hash to a modulated hash function that maps to an empty or positive seesaw counter in SCA. According to the whether the key mapped by the key to insert, the hash modulating can be divided into the following two cases.

In the first case, if the mapped cell is empty in HashModulator (with probability $1 - \mathcal{L}$), there will be at most 2^{η_2} backup modulated hash functions to be used since the ModulatedIndex is η_2-bit in size. Then the probability that the hash modulating is successful, *i.e.*, at least one modulated hash functions map to empty or positive seesaw counter, is given by $P_{ms1} = (1 - \mathcal{L})(1 - \mu^{2^{\eta_2}})$.

In the second case, if the mapped cell is occupied in HashModulator (with probability $\mathcal{L}$), only modulated hash function stored in this cell can be used. Similar to the first case, the probability of successful hash modulating is given by $P_{ms2} = \mathcal{L}(1 - \mu)$.

By combining P_{ms1} and P_{ms2}, the probability of a successful hash modulating is $P_{ms} = \mathcal{L}(1 - \mu) + (1 - \mathcal{L})(1 - \mu^{2^{\eta_2}})$. This completes the proof. □

Lemma V.5. *After inserting N_p positive keys, there will be $\psi \approx 1 - e^{-\frac{k(1 - P_{ms})}{B}}$ ratio of negative seesaw counters with positive cell being non-empty in expectation.*

Proof. When inserting a positive key, the positive cell of a negative seesaw counter is occupied if the hash modulating fails or more than one hash function map the inserted key to negative seesaw counter at the same time during the insertion of one single key. Firstly, as per Lemma V.4, the probability of one unsuccessful modulating is $1 - P_{ms}$. Secondly, we analyze the probability of more than one negative seesaw counter being mapped at the insertion of one single key. Without loss of generality, the applied k hash functions are assumed to be uniformly random, which indicates the number of negative seesaw counters being mapped during one positive key insertion obeys the Binomial distribution $B(k, \mu)$. Therefore, the number of negative seesaw counters mapped during the insertion of one single positive key, say ψ_1 is given by

$$\begin{aligned}
\psi_1 &= \sum_{i=1}^{k} (\mu^i (1 - \mu)^{k-i} ((1 - P_{ms}) \cdot i + P_{ms} \cdot (i - 1))) \\
&= \sum_{i=1}^{k} \mu^i (1 - \mu)^{k-i} (i - i \cdot P_{ms} + i \cdot P_{ms} - P_{ms}) \\
&= \sum_{i=0}^{k} \mu^i (1 - \mu)^{k-i} (i - P_{ms}) - \mu^0 (1 - \mu)^k (0 - P_{ms}) \\
&= \sum_{i=0}^{k} \mu^i (1 - \mu)^{k-i} i - \sum_{i=0}^{k} \mu^i (1 - \mu)^{k-i} P_{ms} + (1 - \mu)^k P_{ms} \\
&= k\mu - P_{ms} + (1 - \mu)^k P_{ms} \\
&= k\mu - P_{ms}(1 - (1 - \mu)^k). \qquad (1)
\end{aligned}$$

By the linearity of expectation, there will be $N_p \cdot \psi_1$ hash functions mapped to negative seesaw counters during the insertion of N_p positive keys. Then, the ratio of negative seesaw counters with positive cells also being occupied is

$$\begin{aligned}\psi &= 1-(1-\frac{1}{m\mu})^{N_p\cdot\psi_1}\\ &= 1-(1-\frac{1}{N_pB\mu})^{N_p\cdot\psi_1}\\ &\approx 1-e^{-\frac{N_p\cdot\psi_1}{N_pB\mu}} = 1-e^{-\frac{k-P_{ms}\frac{1-(1-\mu)^k}{\mu}}{B}}\\ &\approx 1-e^{-\frac{k-P_{ms}\frac{1-(1-k\mu)}{\mu}}{B}} = 1-e^{-\frac{k-P_{ms}k}{B}}\\ &= 1-e^{-\frac{k(1-P_{ms})}{B}}.\end{aligned}$$

This completes the proof. □

Lemma V.6. *After inserting N_p positive keys, $\lambda = 1 - e^{-\frac{k}{B}(1+\frac{\mu}{1-\mu}P_{ms})}$ ratio of non-negative seesaw counters are with non-empty positive cell in expectation.*

Proof. For a given empty seesaw counter in SCA, it is set either if mapped by the initial hash functions or if mapped by modulated hash functions. Meanwhile, during the insertion of N_p positive keys, the number of applied initial hash functions is $(1-\mu)N_pk$ and the number of applied modulated hash functions is $\varphi\cdot P_{ms}$ as per Lemma V.1 and V.4. Therefore, the probability of a non-negative seesaw counter with non-empty positive cell is

$$\begin{aligned}\lambda &= 1-(1-\frac{1}{(1-\mu)m})^{(1-\mu)N_pk+\varphi\cdot P_{ms}}\\ &= 1-(1-\frac{1}{(1-\mu)N_pB})^{(1-\mu)N_pk+\varphi\cdot P_{ms}}\\ &\approx 1-e^{-\frac{k+\frac{(1-(1-\mu)^k)}{1-\mu}P_{ms}}{B}} = 1-e^{-\frac{k+\frac{(1-(1-k\mu))}{1-\mu}P_{ms}}{B}}\\ &\approx 1-e^{-\frac{k+\frac{k\mu}{1-\mu}P_{ms}}{B}} = 1-e^{-\frac{k}{B}(1+\frac{\mu}{1-\mu}P_{ms})}.\end{aligned}$$

This completes the proof. □

C. Cost-weighted FPR of SSCF

Theorem V.1. *When querying a vulnerable negative key, its FPR is $FP_{\mathcal{V}} = \psi^{k-1}(\psi+(1-\psi)k\lambda(1-\mathrm{e}^{-\frac{1}{s}})(1-\mu))$.*

Proof. When querying a vulnerable negative key, it is identified to be positive if it is misidentified with initial hash functions or modulated hash functions. Firstly, with initial hash functions, a false positive occurs when all the mapped seesaw counters are with non-empty positive cells, whose probability is given by $FP_{\mathcal{V}_1} = \psi^k$, where ψ is the ratio of seesaw counters with non-empty negative and positive cell as shown in Lemma V.5. Secondly, if a vulnerable negative key is misidentified with the modulated hash function, there should be only one initial hash function mapping to a seesaw counter with empty positive cell, whose probability is given by $FP_{\mathcal{V}_2} = k\psi^{k-1}(1-\psi)\cdot(\mathcal{L}\lambda)(1-\mu)$. By combining the $FP_{\mathcal{V}_1}$ and $FP_{\mathcal{V}_2}$, we have

$$\begin{aligned}FP_{\mathcal{V}} &= FP_{\mathcal{V}_1}+FP_{\mathcal{V}_2} = \psi^k + k\psi^{k-1}(1-\psi)\cdot(\mathcal{L}\lambda)\\ &= \psi^{k-1}(\psi+(1-\psi)(1-\mu)k\mathcal{L}\lambda)\\ &= \psi^{k-1}(\psi+(1-\psi)(1-\mu)k\lambda(1-\mathrm{e}^{-\frac{1}{s}})).\end{aligned}$$

This completes the proof. □

Theorem V.2. *When querying a non-vulnerable negative key, its FPR is $FP_{\mathcal{N}} = (\mu\psi+(1-\mu)\lambda)^{k-1}((\mu\psi+(1-\mu)\lambda)+(1-(\mu\psi+(1-\mu)\lambda))k(1-\mu)\lambda(1-\mathrm{e}^{-\frac{1}{s}}))$.*

Proof. Similar to that of querying a vulnerable negative key, a non-vulnerable negative key is misidentified if it is either misidentified with initial or modulated hash functions. For the initial hash functions, a false positive occurs when all mapped cells are occupied, which has probability $FP_{\mathcal{N}_1} = (\mu\psi+(1-\mu)\lambda)^k$. As for the modulated hash functions, its probability is $FP_{\mathcal{N}_2} = k(\mu\psi+(1-\mu)\lambda)^{k-1}(1-(\mu\psi+(1-\mu)\lambda))\mathcal{L}(1-\mu)\lambda$. Therefore, the FPR of querying non-vulnerable negative keys can be derived as

$$\begin{aligned}FP_{\mathcal{N}} &= FP_{\mathcal{N}_1}+FP_{\mathcal{N}_2}\\ &= (\mu\psi+(1-\mu)\lambda)^{k-1}((\mu\psi+(1-\mu)\lambda)\\ &\quad+(1-(\mu\psi+(1-\mu)\lambda))\cdot k(1-\mu)\lambda\mathcal{L}))\\ &= (\mu\psi+(1-\mu)\lambda)^{k-1}((\mu\psi+(1-\mu)\lambda)\\ &\quad+(1-(\mu\psi+(1-\mu)\lambda))\cdot k(1-\mu)\lambda(1-\mathrm{e}^{-\frac{1}{s}})))\end{aligned}$$

This completes the proof. □

Theorem V.3. *Suppose the encoded N_n negative keys accounting for ϵ ratio of all costs, then cost-weighted FPR of SSCF is $CFPR(B,s) = \epsilon\cdot FP_{\mathcal{V}}+(1-\epsilon)\cdot FP_{\mathcal{N}}$, where B is the seesaw counters per key in SCA and s is the scale factor of HashModulator.*

Proof. The cost-weighted FPR is composed of two parts: the false positive of vulnerable negative keys and non-vulnerable keys. For a given vulnerable negative key, its FPR is given by FP_V . By the linearity of expectation, the cost-weighted FPR arising from vulnerable keys is $\epsilon\cdot FP_{\mathcal{V}}$, where ϵ is the ratio of costs of the encoded vulnerable negative keys. Similarly, as per Theorem V.2, the cost-weighted FPR of non-vulnerable key is $(1-\epsilon)\cdot FP_{\mathcal{N}}$. Therefore, the cost-weighted FPR of SSCF is $\epsilon\cdot FP_{\mathcal{V}}+(1-\epsilon)\cdot FP_{\mathcal{N}}$. □

D. Parameter Optimization

Suppose the memory space budget is M, the HashModulator is allocated with $\alpha\cdot M$ space and SCA is allocated with $(1-\alpha)M$ ($\alpha\in[0,1]$). Meanwhile, there are m seesaw counters in SCA of SSCF and each seesaw counter is composed of two fields, including θ_1-bit negative cell and θ_2-bit positive cell. Besides, the positive key capacity of SSCF is N_p and each positive key is allocated with B seesaw counters cells. Therefore, we have $(1-\alpha)M = m\cdot(\theta_1+\theta_2) = N_p\cdot B\cdot(\theta_1+\theta_2)$. Then α can be formulated as

$$\alpha = 1-\frac{BN_p(\theta_1+\theta_2)}{M} \tag{2}$$

Concerning HashModulator, it is allocated with $(1-\alpha)M$ space with n cells, each of which has two fields: η_1-bit *ModulatedCounter* and η_2-bit *ModulatedIndex*. As per Lemma V.3, we can reformulate n as:

$$n = sN_p(1 - e^{-\frac{k^2}{B}\frac{N_n}{N_p}}) \approx \frac{s \cdot k^2 \cdot N_n}{B}. \tag{3}$$

Besides, the space of HashModulator is αM, which consists of n ($(\eta_1 + \eta_2)$-bit in size) cells of size. Therefore, we have $\alpha M = n(\eta_1 + \eta_2)$ and s can be reformulated as $s = \frac{\alpha M}{(\eta_1+\eta_2)(\frac{k^2 \cdot N_n}{B})} = \frac{MB - B^2 N_p(\theta_1+\theta_2)}{(\eta_1+\eta_2)(k^2 \cdot N_n)}$. Finally, as per Theorem V.3, the problem of how to obtain an optimal cost-weighted FPR regarding space ratio allocated to HashModulator can be formulated as follows:

$$\begin{aligned} \min \quad & CFPR(B, s) = \epsilon \cdot FP_{\mathcal{V}} + (1-\epsilon) \cdot FP_{\mathcal{N}} \\ s.t. \quad & s {=} \tfrac{MB - B^2 N_p(\theta_1+\theta_2)}{(\eta_1+\eta_2)(k^2 \cdot N_n)}, \quad B \in [1, \tfrac{M}{(\theta_1+\theta_2)N_p}]. \end{aligned} \tag{4}$$

Note that similar to standard Counting Bloom filter, $\theta_1 = 1$ and $\theta_2 = 4$ by default in this paper. How to set k, η_1, η_2 is discussed in our evaluations and we focus on how to obtain an optimized B here. Considering that Equation 4 can be evaluated with $O(1)$ time complexity, the parameter B is an integer within range $(1, \frac{M}{(\theta_1+\theta_2)N_p})$. Therefore, the optimal B can be found within $O(log_2^{\frac{M}{(\theta_1+\theta_2)N_p}})$ time with binary search. Finally, with the obtained optimal B, the optimal ratio α of space allocated to HashModulator can be derived by plugging B into Equation 2.