WHAT IS CLAIMED IS:

1. A method comprising:

receiving, by a computing device, an input data stream having a plurality of input elements;

for each of the input elements received, determining whether the input element satisfies a filtering condition;

for each of the input elements received that satisfies the filtering condition, performing a write operation to store the input element in a memory subsystem;

for those of the input elements received that do not satisfy the filtering condition, performing at least a dummy write operation to the memory subsystem; and

evicting the memory subsystem to an output data stream when the memory subsystem is full.

2. The method of claim 1, wherein the write operation includes concatenating a positive indicator bit with the input element.

3. The method of claim 2, wherein the write operation further includes encrypting a result of the concatenating, and storing the encrypted result in the memory subsystem.

4. The method of claim 1, wherein ~~a dummy~~ the dummy write operation is performed for each of the input elements received that does not satisfy the filtering condition.

5. The method of claim 1, wherein the dummy write operation includes concatenating a negative indicator bit with a data string having a same length as an input element.

6. The method of ~~claim 4~~claim 5, wherein the dummy write operation includes encrypting a result of the concatenating, and storing the encrypted result in the memory subsystem.

7. The method of ~~claim 4~~claim 5, wherein the data string is a string of zeros.

8. The method of claim 1, wherein the memory subsystem includes a trusted memory and an unprotected memory.

9. The method of claim 8, wherein at least one of the input elements that satisfies the filtering condition is stored in the trusted memory.

10. The method of claim 8, wherein at least one of the input elements that satisfies the filtering condition is stored in the unprotected memory.

11. The method of claim 8, wherein the dummy write operation writes to a designated location in the memory subsystem.

12. A computing system comprising:

a memory subsystem;

a processor; and

a non-transitory computer readable storage medium storing code, which when executed by the processor, cause the processor to perform operations including:

receiving an input data stream having a plurality of input elements;

for each of the input elements received, determining whether the input element satisfies a filtering condition;

for each of the input elements received that satisfies the filtering condition, performing a write operation to store the input element in ~~a memory~~ the memory subsystem;

for those of the input elements received that do not satisfy the filtering condition, performing at least a dummy write operation to the memory subsystem; and

evicting the memory subsystem to an output data stream when the memory subsystem is full.

13. The computing system of claim 12, wherein the write operation includes concatenating a~~n~~ positive indicator bit with the input element, encrypting a result of the concatenating, and storing the encrypted result in the memory subsystem.

14. The computing system of claim 12, wherein ~~a dummy~~ the dummy write operation is performed for each of the input elements received that does not satisfy the filtering condition.

15. The computing system of claim 12, wherein the dummy write operation includes concatenating a negative indicator bit with a data string having a same length as an input element, encrypting a result of the concatenating, and storing the encrypted result in the memory subsystem.

16. The computing system of claim 15, wherein the data string is a string of zeros.

17. The computing system of claim 12, wherein the memory subsystem includes a trusted memory and ~~a~~ an unprotected memory.

18. The computing system of claim 17, wherein at least one of the input elements that satisfies the filtering condition is stored in the trusted memory.

19. The computing system of claim 17, wherein at least one of the input elements that satisfies the filtering condition is stored in the unprotected memory.

20. The computing system of claim 17, wherein the dummy write operation writes to a designated location in the memory subsystem.

21. The computing system of claim 17, wherein the trusted memory is of a size that is dependent on a length of the input element, and a logarithmic of a size of the unprotected memory.

22. A method comprising:

receiving, by a computing device, a first input element of an input data stream having a plurality of input elements;

determining that the first input element satisfies a filtering condition;

performing a write operation to store the first input element in a memory subsystem;

receiving a second input element of the input data stream;

determining that the second input element fails the filtering condition;

performing a dummy write operation to a designated location in an unprotected memory of the memory subsystem; and

evicting the memory subsystem to an output data stream when the memory subsystem is full.

23. The method of claim 22, wherein the first input element is stored in a trusted memory of the memory subsystem.

24. The method of claim 22, wherein the second input element is stored in the unprotected memory of the memory subsystem.

25. The method of claim 22, wherein the memory subsystem further includes a trusted memory having a size that is dependent on a length of the input elements of the input data stream, and a logarithmic of a size of the unprotected memory.