Chapter 5
Conclusion

In this thesis, a functional and scalable custom automata system has been proposed for modern IDS system. The newly Snort rules can be processed with much less automata data size while overcoming the disadvantages of traditional pattern-matching algorithms. The works of processing pattern-matching, pattern-relationships, and PCRE patterns can now be done by the proposed automata instead of by extra CPU computing. As the size of constructed automata is very tiny, it is a very suitable for the design of a hardware parallel pattern matching engine of the IDS.

Some experiments have been performed to compare the performance and the data structure size between the proposed custom automata and the open Snort system. Our custom automata system has perfect balance among functionality, performance, and data size. Based on this new design, all the header fields and content keywords are treated as patterns for matching, and this provides a more easy way to design an IDS or related network device.

From the experimental results, we found that the data size of our custom automata system is much less that that of Snort. Moreover, for Snort system, it is necessary to handle complex rules and pattern-relationship in post-processor, but for our custom system, all these can be done in automata while doing matching. Although the software simulation indicated that the performance of the proposed automata system is worse than that of Snort, the proposed automata system can easily be implemented in hardware (FPGA or ASIC) to achieve much better performance. Another advantage of the proposed automata system is the ability to handle various

types of rules, especially in Regular Expression and pattern-relationship. Most importantly, with the powerful multi-pattern matching kernel, the custom automata system is very flexible to face the future complex rules. We can expect that the IDS or firewall rules will become more complex in the future for managing more emerging applications. But the only thing we need to do is to rewrite a suitable parser to translate the rules into the format accepted by the custom automata. With the capability to process Regular Expression and pattern-relationship, along with the efficient parallel hardware architecture, the proposed custom automata system is very suitable for the designing the next generation IDS or firewalls.

In addition to the custom automata system, a Regular Expression automata for multi-pattern matching has also been implemented. This can speedup the matching time especially for those input rules mixed with normal patterns and Regular Expressions. The Regular Expression automata is used to process those special patterns, and hash filter and matching engine are employed to achieve multi-pattern matching in hardware. With M hash functions, the hash filter could offload the matching engine by dropping those patterns whose hash values are not matched. Using N P-bit ALUs to do multi-pattern matching is a good solution of implementing in network ASIC or network processor. Also the most advantage is that the data size of the proposed custom automata system is much less than that of Snort and we could put the whole automata into SRAM.

There are some things we could optimize in the future. Although header automata usually finishes matching process before content automata does, there are more and more short packets now, especially in P2P network traffic. We may also implement header automata in parallel hardware architecture or rewrite all the header rules into Regular Expression form with our Regular Expression automata. Besides, in our hardware design, we use N P-bit ALUs in matching engine. It may make the

performance better or worse by choosing different values of *N* and *P* with the assumption that *N*P* is greater than the length of every pattern in Snort rules. Finally, it may be a good idea to separate our automata into multi-level automata, like the meaning of “automata in automata” discussed in Chapter 4. The memory requirements may become more but the performance of our system may be better.

Using automata in IDS is a good solution but it is not easy to take the perfect balance between performance and data size. Our custom automata system in hardware has low memory requirements with the special data structure of automata and more functionality with the ability to handle pattern relationships and Regular Expressions. It also achieves the multi-pattern matching and the performance is nice because of the design of hash filter and parallel ALUs in matching engine. In brief, the more complex intrusion rules appeared in this modern world, the more suitable it is as a good choice to use the proposed custom automata system to detect and defeat those intrusions.