Summary Part 1
HIDE: An Infrastructure for Efficiently Protecting Information Leakage on the Address Bus

Mrinal Nath (ID: 3307043)
April 11, 2005

What are the problems solved by this paper?
XOM-based architecture was proposed to provide copy- and tamper-protection in hardware so that the software can not be ‘cracked’ and copied illegally (IP rights violations). But XOM protects only the data and the actual instructions; it does not protect the addresses on the address bus. Unfortunately, the sequence of addresses on the address bus can be used to infer the control-flow information of the program, which can be used to ‘crack’ it. Efforts to ‘hide’ the addresses have been largely unsuccessful so far. This paper proposes a technique to hide the addresses while minimizing the performance overheads.

What are the approaches attempted by this paper?
The basic idea is that if the actual address sequence presented on the address bus is totally randomized (probabilistically fixed distribution), then anyone snooping the address bus will not be able to make much sense of the sequence of addresses. Towards this goal, the paper proposes microarchitectural modifications like HIDE caches which support block locking and permutation operations. In addition, they suggest some compiler optimizations for code and data layout and other runtime optimizations to reduce overheads and improve the level of security.

What are the main conclusions of this paper?
HIDE is a very effective and (most importantly) low-overhead technique to provide substantial levels of security for the address sequences on the address bus. The hardware additions are quite small in size, so the increase in area and power would not be very large. The overall performance degradation is only around 1.5% and the increase in bus traffic due to the permutations is small.