Lecture Note

Chapter 7:
Multiprocessor and Many-Core Protections

CYENG 225: Microcontroller Essentials for Cyber Applications

Instructor: Dr. Shayan (Sean) Taheri

Gannon University (GU)

Chapter 7 Overview

- **Major Items**
  - It discusses security issues in the design of secure processor architectures for computer systems which have more than one processor chip, or systems which have many processor cores on one chip.
  - It focuses on shared memory multiprocessors and distributed shared memory systems.
  - It covers security issues in many-core processor designs.
  - It provides list of assumptions for multiprocessor and many-core secure architectures.

Security Challenges of Multiprocessors and Many-Core Systems

- The overarching theme of the multi-processor designs is the communication.
- Uni-processor designs mainly focus on protection of the off-chip communication going to and from memory.
- The memory does not usually initiate communication so it is considered a passive element.
- responds to requests from the processor. In multi-processor there is now a new dimension which is the processor to processor communication.
- In multi-processor designs, securing inter-processor communication is a new challenge.
- In multi-processor designs, multiple processors, each on physically separate chip, are connected together, and connected to the memory.
- The interconnect is assumed to be more easily probed than chips themselves, thus motivating the need to secure the communication, and any data, that is transferred between different chips.
- Multi-processor secure designs need to ensure confidentiality, integrity, and authenticity of the messages going among the processor chips, and memory as well.
- One or more of the processors could be malicious (it is rather easy to swap out processors on the motherboard), thus some mechanisms are needed for authentication of the chips.

Security Challenges of Multiprocessors and Many-Core Systems (Cont.)

- In many-core designs, the threats that multi-processor designs have worried about on the outside of the chip, have now moved into the processor chip.
- Many-core designs increase performance by including more and more processing cores within a processor chip (rather than connecting many chips together).
- Thanks to continued advances in manufacturing technology, tens or hundreds of processor cores can fit on a single chip today.
- Such expansion of cores within processor chip can, however, create a new threats: where one or more of the cores in untrusted.
- As in the multi-processor design, there are now many entities communicating, and the communication needs to be protected.

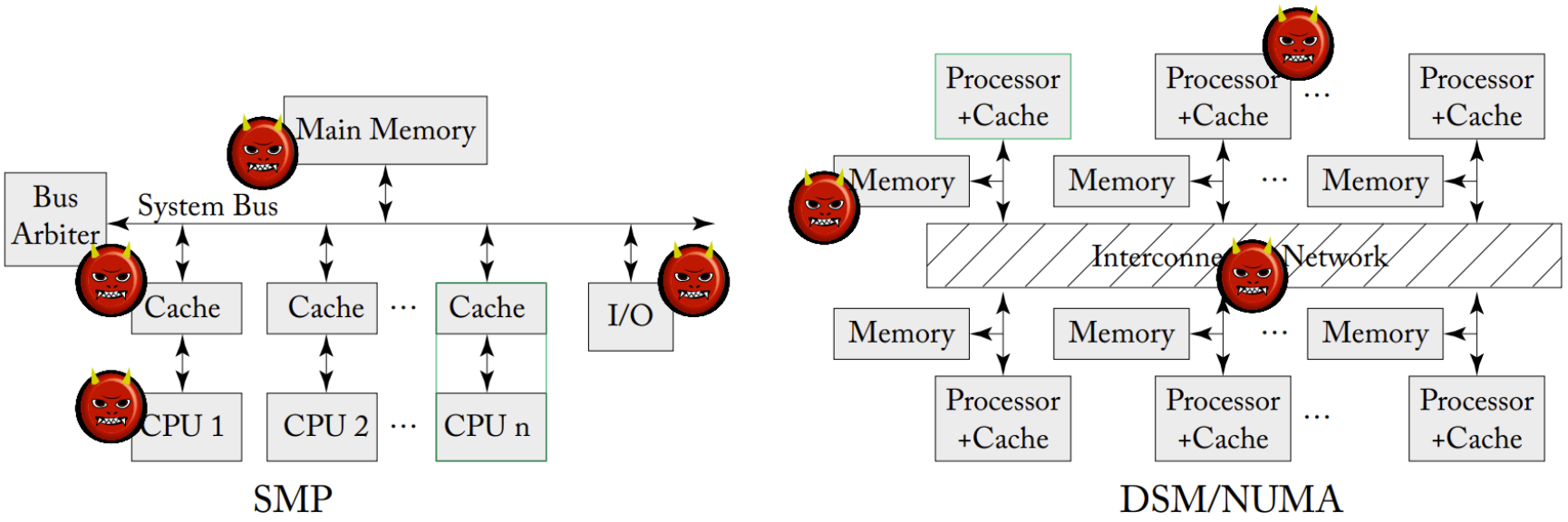
Multiprocessor Security

- Symmetric Multi Processing (SMP) and Distributed Share Memory (DSM) are two main design types for Multiprocessor systems.
- In SMP, there is typically a single memory bus which is connected to all the processors, and which is also connected to the main memory.
- All processors in SMP connect to the same main memory.
- In DSM, there is an interconnection network that connects all the processors. → As the name implies, the memory is distributed.
- In DSM, each processor has some memory that is physically near it, and thus is fast to access, and the rest of the memory is physically far (as it is near one of the other processors) and it is slow to access.

SMP and DSM Threat Model

- SMP and DSM, also referred to as Non-Uniform Memory Access (NUMA), offer two ways of connecting many CPUs together.
- From the perspective of a processor chip, all other processor chips, memories, or I/O components can be untrusted.
- Unlike with uni-processor designs, one danger of SMP or DSM designs is that one or more of the processors could be malicious.
- The different attacks can come from the untrusted software or hardware, especially, again, other processors can be potentially malicious.

Multiprocessor Security and SMP and DSM Threat Model



Block diagram of a typical Symmetric Memory Multiprocessor (SMP) system and a typical Distributed Share Memory (DSM). The usual threat model assumes the processor is trusted (highlighted in green) but the other components are untrusted, especially other processors can be sources of attacks.

Symmetric Memory Multiprocessor Security

- In SMP designs, there is typically a single bus which is connected to the processors.
- The memory is also connected to this bus.
- All communication is done through the shared bus.
- The shared bus is typically part of the motherboard, and may be more easily probed than the processor chips themselves.
- Consequently, secure SMP designs need to protect the communication on the shared bus, plus potentially give protection from one or more of the other processors.
- They also need to protect memory as in uni-processor designs.

Symmetric Memory Multiprocessor Security (Cont.)

- Confidentiality and Integrity of Communication.
  - Work on secure SMP points to a number of solutions which can be leveraged to protect the communication among the different processors.
  - Due to performance constraints, counter mode AES is the typical choice for encryption.
  - The counter values can be pre-generated ahead of time, and only the resulting pads need to be XORed with the data, adding minimal latency.
  - The challenge is to keep track of the counters for the communication among the processors.
  - One solution is to have each source-destination pair of processors use a dedicated set of counters for encryption of the data traffic.
  - For integrity protection, message authentication codes (MACs) can be used.
  - Each sender can create a MAC for authentication and integrity checking of the message.
  - As a performance improvement, AES in Galois/Counter Mode (GCM) combines both encryption and authentication.
  - Use of the so-called authenticated encryption mode of a symmetric cipher allows the system to generate the MAC at the same time as the message is being encrypted.
  - Confidentiality and integrity protections assume each source-destination pair of processors shares a secret key that can be used to do the encryption and MACs.

Symmetric Memory Multiprocessor Security (Cont.)

- Confidentiality and Integrity of Memory.
  - SMP systems require memory protection just as uni-processor systems.
  - Memory protection can be done using any of the variants of the Merkle trees.
  - As the memory is a shared resource, all the processors need to keep track of the memory updates, and possibly explicitly share the root node of the integrity tree—or individually update the tree root, but at all times they need to have a consensus about what is the correct tree root value.
  - An advantage of SMP systems is that each processor sees all the messages sent on the memory bus.
  - Each processor can snoop on the memory bus and when it sees a message going to the memory it can authenticate the message.
  - An alternate solution is to maintain per-processor memory integrity trees.
  - Each processor can maintain an integrity tree for the memory it is working with.
  - Data shared between processors can be put in physical memory that is not being checked for integrity (so after one processor updates it, the other can read it without encountering a validation error in its integrity tree).

Symmetric Memory Multiprocessor Security (Cont.)

- Memory Access Pattern Protection.
  - Because in simultaneous multithreading (SMT) every processor can observe all memory accesses, attacks based on traffic analysis are more easy (e.g., compared to uni-processor setup where the attacker has to physically probe the memory bus, here one of the SMP processors could simply be malicious and it is directly plugged into the bus).
  - One counter-measure is the ORAM which can hide access patterns.
  - Each processor in SMP would have to include an instance of ORAM to hide its access patterns from others.
  - A disadvantage is that this may increase reads and writes on the memory bus, leading to more integrity related operations.

Symmetric Memory Multiprocessor Security (Cont.)

- Key Management.
  - Each processor in an SMP system needs to have its own key, from which keys for confidentiality and integrity checking can be derived.
  - The key can either be burned in by the manufacturer, or PUF can be used to derive the key.
  - The challenge is that the other processors in the SMP system need to be informed of the legitimate keys of all the other processors.
  - How to efficiently implement the key sharing among SMP nodes is open problem.
  - The keys could be loaded into the processors at boot time by some trusted mechanism.
  - Storage of the shared secrets in each processor needs attention as well.
  - An attacker could swap out a processor, try to extract the key, and then insert a malicious processor (but now with the right shared keys) into the socket.
  - Thus, keys stored on the processors should be writeable (to initialize them), but only readable by the hardware (so there is no easy way to read out the key, short of invasive physical attack).

Distributed Shared Memory Security

- In DSM designs there is no longer a single bus that all processors can snoop on.
- Rather, processors may be connected in a mesh or other network.
- The connections are typically point-to-point, so all processor chips do not see all the traffic, rather data packets take different routes from source to destination.
- This allows many packets to be in flight at same time and improves performance compared to a shared bus in SMP.
- In NUMA the memory itself is also distributed, each processor is associated with a piece of memory that is physically co-located with that processor.
- Memory accesses are non-uniform, accessing local memory is faster than memory further away, allowing system to have lots of memory, with the benefit that data stored in memory close to the processors can be accessed faster.
- In addition, DSM systems have coherency protocols to keep track of which processor has the latest copy of which data.
- Thus, there are the coherency messages (a memory access on one processor may trigger a number of coherency messages needed to bring in latest copy of the data into that processor).

Distributed Shared Memory Security (Cont.)

- Confidentiality and Integrity Protection of Communication and Memory.
  - For processor-to-memory communication, confidentiality can be again ensured with a block cipher such as AES in counter mode, as it offers best performance (if the pads are pre-computed correctly and in time).
  - All data's confidentiality could be protected with the symmetric key encryption.
  - For integrity MACs can be used, or the AES GCM mode can combine both confidentiality and integrity; and Merkle tree can be used for checking the integrity of the main memory.
  - For processor-to-processor communication, encryption and MACs can be used.
  - The plethora of cache coherence messages, makes it prohibitive to encrypt and hash all the messages to ensure security.
  - One solution, proposed in a secure DSM design, is to only ensure that attacks that do not result in coherence protocol anomalies can are detectable.

Distributed Shared Memory Security (Cont.)

- Access Pattern Protection of Memory.
  - In DSM all processors do not get to observe all the memory accesses or all the coherence messages.
  - This can be leveraged to simplify memory access obfuscation.
  - Routing of some packets could be controlled in a special way, or randomized, to hide the access patterns of one processor form the others.
  - This may even remove the need for access pattern obfuscation all together.

Distributed Shared Memory Security (Cont.)

- Key Management.
  - As in SMT, key distribution among the processors that are part of the DSM system is a challenge.
  - Two main options are to pre-install the other processors' keys, or use public-key cryptography for processors to get each other's keys.
  - Overhead of public key cryptography is usually prohibitive.
  - Alternatively, and depending on the threat model, is to leverage the interconnect.
  - One advantage of DSM is that there are point-to-point links, so two processors can communicate directly without others seeing the messages.
  - This can be used to distribute pair-wise keys, where only the sender and receiver knows what the key is (assuming there are no external attackers).
  - As in SMT, an attacker may be able to extract the keys, and he or she can potentially insert a malicious processor into one of the sockets and impersonate a correct one.
  - Keys need to be protected from read out by any untrusted software (potentially even untrusted operating system or hypervisor).

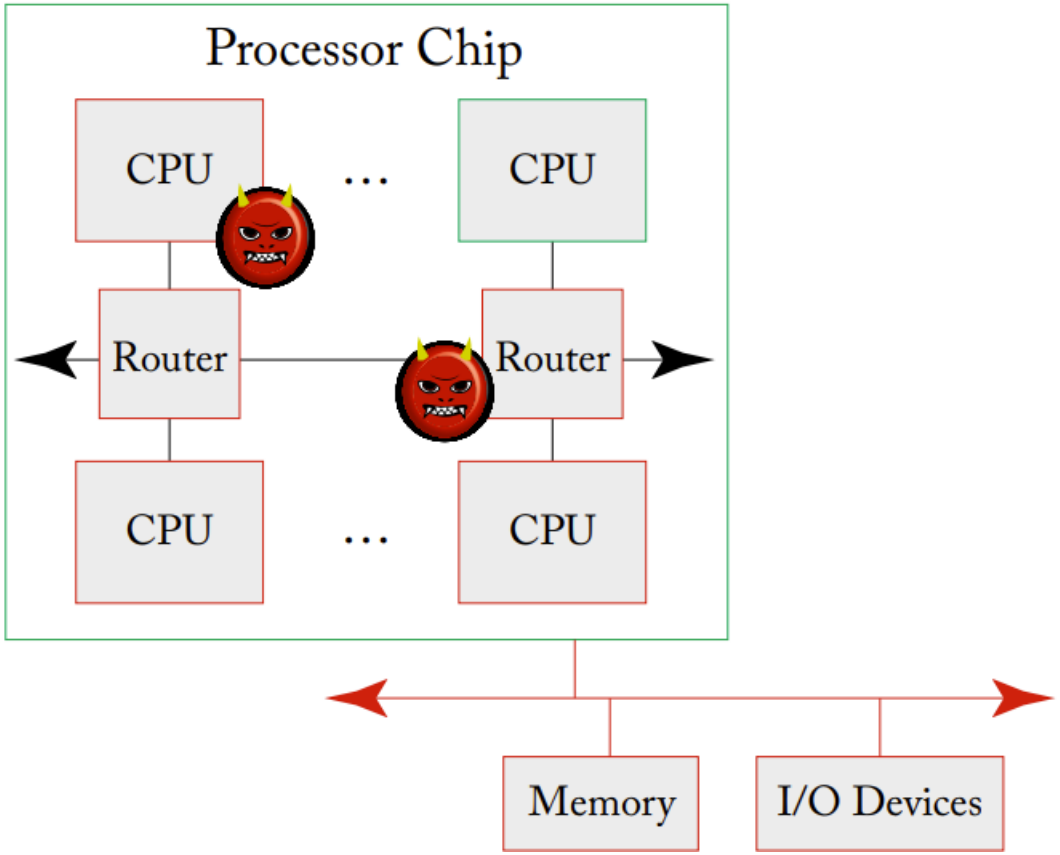
SMP and DSM Tradeoffs

- While SMT is becoming an older design style, with most processors moving to DSM, one benefit of SMT is that encryption and other cryptographic computations can be done fast enough in relation to (the slower) SMT data rates.
- This allows for encryption and integrity checks being done on reads and writes by each processor.
- Efficient key sharing among SMP processors is an open-issue today.
- While optimizations such as separate integrity trees can potentially improve performance of the system compared to a single tree.
- Point-to-point links in DSM can be leveraged for security, as not all processors see all the communication and messages.
- They can also aid in sharing keys among the processors.
- Challenges of DSM include rather fast interconnect, necessitating either very fast cryptographic modules, or judicious protections of only certain messages passed between processors chips.

Many-Core Processors and Multi-Processor System-on-a-Chip

- Many-core processors and multi-processor system on a chip (MPSoC) designs share the characteristic of having many processing cores on the same chip.
- Similar to SMT or DSM, there are many processing elements, but instead of these processors being on separate chips, and the interconnect being exposed on the motherboard, they are all inside the same chip.
- In this configuration, the memory is typically a separate chip. However, with recent trends in 3D integration the memory may also be in same package, or in a package-on-package (PoP) configuration.
- Typically, many-core processors have same type of a processor core, while MPSoC com bine various cores, often from different manufacturers.
- In both cases, the cores are connected by a network-on-a-chip (NoC).
- The are three main components of the NoC: the processors, the routers that are used to route traffic between processors, and the wires carrying the data.

Many-Core Processors and Multi-Processor System-on-a-Chip (Cont.)

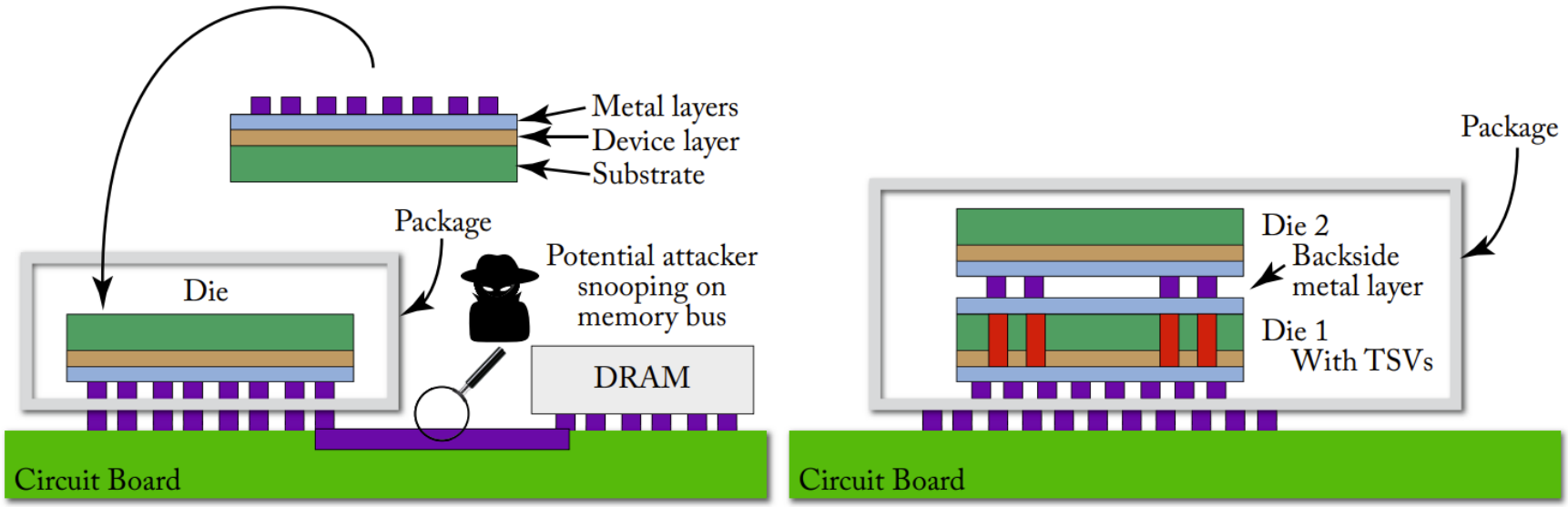


Potential threats agains many-core systems include untrusted processors, routers, or interconnect between the routers. Off-chip components such as the main memory or I/O devices are untrusted as well.

3D Integration Considerations

- There is a recent trend toward 3D integration (variably called either 3D or 2.5D).
- Especially with MPSoC systems, used in small embedded device, now commonly called Internet-of-Things, the main DRAM memory is integrated in the same package as the MPSoC, or in a package-on-package (PoP) configuration.
- This has notable performance benefit as the wires between processor chip and memory are shorter.
- Also, the vertical 3D integration can accommodate higher-bandwidth communication.
- The security advantage of the 3D or 2.5D integration is that the wires going between the processor and the memory are more difficult to access (they are either in the package, or sandwiched between two packages in the PoP configuration).
- This physical aspect of the design can be used to formulate a threat model where external attacks on processor-to-memory interconnect are not considered, removing the need to do encryption, hashing, or access pattern protection.
- A related trend is the embedded DRAM (eDRAM).
- Embedded DRAM is dynamic random-access memory integrated on the same die as the main processor. It has the same potential benefit as 3D integration in that processor and memory connections are not exposed outside the processor package.

3D Integration Considerations (Cont.)



Example of traditional 2D design (left) and 3D integration (right). With the 2D design, processor and memory interconnect wires are exposed on the circuit board, allowing for potential attackers to snoop or invasively attack the. With the 3D integration, all wires going to or from memory are in the processor package.

Multiprocessor and Many-Core Protections Assumption

- In addition to the existing uni-processor assumptions, designs with multiple processors or cores assume that the inter-processor communication will be protected.
- Confidentiality needs to be ensured such that if two processors are communicating, other processors cannot snoop on the communication.
- Integrity needs to be ensured such that one processor cannot modify memory of another processor.
- Communication pattern protection also needs to be ensured as malicious processor can observer the communication of other processors.
- Furthermore, the protected inter-processor communication assumption requires that different processors be able to mutually authenticate each other.

Assignment

- **Reading Assignment:**
  - Zferer, J., 2018. **Principles of secure processor architecture design**, ser. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 9048, pp.1-175.
    - ✓"Chapter 7: Multiprocessor and Many-Core Protections", Pages 75-84.

Questions?