NoC security using multipath routing

Radu Stefan
Computer Engineering
Delft Institute of Technology
Email: R.A.Stefan@tudelft.nl

Kees Goossens
Computer Engineering, Delft Institute of Technology
NXP Semiconductors
Email: K.Goossens@ewi.tudelft.nl

***Abstract*—Title: Network-on-chip security using multipath routing**

Authors: Radu Stefan, Kees Goossens

keywords: noc, multipath, security

In a highly competitive consumer electronics industry, companies try to protect their IP and that of their customers from reverse-engineering attempts and sometimes they attempt to crate barriers against unauthorized use of their products in applications like Digital Rights Management or DRM. This task is made more difficult by the fact that the devices needing protection may be owned by the attackers themselves and can be physically probed. Technology scaling improves chip security by increasing the cost and technological requirements of an attacker, however several components like the interconnect, be it a conventional bus-based or a network-on-chip, remain susceptible to attacks as they usually employ long wires in the upper metal layers. In this study we propose a method of improving chip-level security by using multipath routing in networks-on-chip. The technique consists of splitting a communication channel over multiple paths in order to make it harder to intercept. The selection of paths can be either deterministic, known at design time, or indeterministic, for example based a hardware number generator. In both cases the technique is transparent for the user of the communication channel, which does not need any changes in terms of hardware, while only minimal additional hardware is required for the network interfaces.

I. Introduction

As technology became prevalent in the modern society, electronic identification techniques started to replace traditional methods.

Although the methods changed, the basic principle remains the same, the user presents a unique token as proof for obtaining access to a specific location of piece of information, for authorizing payment or getting access to subscription based services.

Some tokens contain security elements restricting their use to their possession by a single person, for example the photos on traditional id cards, or the equivalent electronically signed form, but this approach is not always possible as some services can be provided remotely, for example over the phone or the Internet, and the physical presence of a person cannot be verified. Furthermore, in some situations, a feedback channel is not available for the distributors of content, TV broadcast companies and TV cable operators have no way of knowing in general which or how many users are watching their programs. A common approach in such situation is to encode the broadcasted content and make the decoding possible only through the possession of a physical token, in this case a decoding circuit. This solution also provides anonymity to the receivers, as the tokens do not need to be linked to the identity of the person.

The basic assumption for the proper functioning of this system is that the tokens or their functionality cannot be replicated. This was ensured initially by the fact that users lacked the technological facilities to perform this replication, although the technology itself was not out of reach even for the budget of private individuals (just to give an example, copying the magnetic strip found in many access cards). One may argue that it is not the technological difficulty, but the threat of legal retaliation that is limiting this type of behavior.

A further refinement of the technology was to include integrated circuits in these identification tokens, the so-called “smart cards”, or directly into the appliances used in the distribution of content. Telephone sim-cards, newer credit cards, and video players starting with the DVD generation are good examples of this approach.

While this technique effectively places the replication technology out of the reach of average individuals it does not present a guarantee against highly motivated individuals or organizations with substantially higher budgets. Several lines of attack were suggested against this approach to security and the response was a further improvement of the technology, defending against these attacks.

Timing attacks

This is probably the most simple type of non-invasive (the device can be studied from the outside, without deterioration and without the device being able to detect the attack) cipher breaking techniques, and it consists of analyzing small changes in the duration of running the cryptographic algorithm on the targeted device when presented with different inputs. Although in practice the problem is more complex, a simple example would be that of a loop checking an input password against the correct version which is stored on chip. If the loop terminates immediately when a wrong character is found the attacker can determine how many characters were guessed correctly. Real cryptographic algorithms are also vulnerable as they frequently employ large number arithmetic operations whose duration may depend on the value of the operands. Possible lines of defense are modifications to the algorithm, possibly requiring it to performed unneeded operations, either forcing it to run in constant time, or randomizing the running time.

Power analysis

A second non-invasive technique consists of monitoring the power consumption of the circuit when presented with various input values. The power profile may change as the chip is performing different operations, or it may change based on the values of the operands as different state transitions consume different amounts of energy. Logic synthesis backends can translate the design into structures insensitive to the direction or presence of transitions [1], [2]. Inserting circuits with random power consumption is also a possibility, although an attacker may be able to find and disable them.

Electromagnetic Analysis

Consists of measuring the electromagnetic radiation emitted by a device or the variation of the electromagnetic signature across multiple runs with the purpose of revealing information about the internal functioning of the device and possibly stored cryptographic keys. Shielding and addition of noise may be possible lines of defense.

Semi-invasive techniques

Proposed by [3] employ the removal of the packaging of the chip, but do not require electrical contact with the actual circuit (the passivation layer is thus preserved and the chip can continue normal functioning). Optical methods can be used to alter the chip functionality (for example examine or toggle the state of memory cells), possibly disabling some of the countermeasures, or causing errors in the cryptographic algorithms which would result in the disclosure of keys.

Physical probing

An attacker can potentially use logical probes directly on the internal chip wiring. Long wires belonging to the interconnect are particularly susceptible to this type of attack. Reverse engineering of the chip is a possibility. Countermeasures consist of sensors that would disrupt the normal functioning of the chip when tampering is detected.

Physical modifications to the chip

It may be possible to selectively damage security elements on the chip thus enabling one of the previous techniques. Vias can be created using ion implantation to connect to wiring in the lower metal layers.

Other techniques

May consist of unconventional methods like irradiating or freezing the chip (DRAM cells were shown to preserve their contents without refresh at low temperatures) with the purpose of obtaining a snapshot of the memory contents that can then be retrieved trough other methods.

In our study we target improvements in chip-level security specifically against Physical Probing attacks, but to some extent also against Electromagnetic and Power Analysis, by making the attacks more difficult and randomizing the behavior of the chip. The interconnect between different modules on chip is particularly sensitive to attacks as it consists of long lines in the upper metal layers which can be easily probed.

Our technique consists of alternating or randomizing the path used for sending the data over a network-on-chip. We consider two options, one of using a static schedule for the path selection and one of leaving this task to a random number generator. Although the second option has the advantage of introducing additional randomness in the chip behavior it also has the disadvantage that more resources need to be allocated for the same communication channel.

II. Proposed architecture

We use as a base for our proposal the existing implementation of the Æthereal network [4]. The Æthereal network provides connection-based communication, either explicit in the form of data streams or offering memory access semantics which allow IPs to transparently perform read and write operations to remote memories. Locally, the IP is connected to a Network Interface shell using a typical bus protocol (Figure 1). For a secure implementation, this connection should not span large distances and can in practice be a point-to-point connection. The network interface shell has the role of encoding the bus transaction request into a request message organized into words matching the network link width.

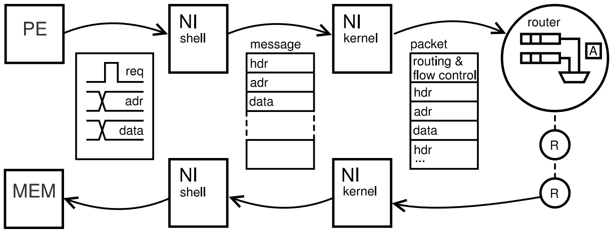

Fig. 1. Architecture of a system based on the Æthereal network .

A network interface kernel is responsible for introducing the packet into the network, providing the necessary routing headers and ensuring the end-to-end flow control. A schedule computed at design time ensures that inside the network, packets travel without contention and without collisions "Contention-free routing" [4]. At destination, an NI shell converts again the message into a bus transaction which is then served by a slave device. The response follows the same sequence of transformations on the return path.

Figure 2 illustrates the suggested implementation. The IP, NI shell and kernel and one local router should be clustered inside a single processing tile and should only be connected using short links, otherwise an attacker could probe any of these connections, defeating the mechanisms proposed in this work. Multipath routing cannot be of use here as there is one single link connecting each pair of these elements.

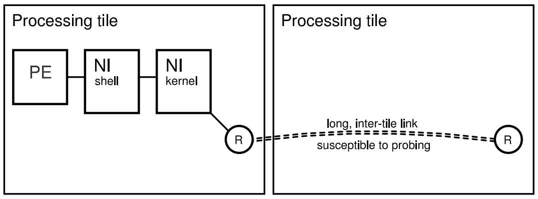

Fig. 2. Long vs. short links .

After the first router packets have a choice of multiple paths to follow to the destination. The paths are computed at design time. The paths are computed in such a way that they are always disjoint, that is, there is no single link that would provide all the packets belonging to one connection if monitored by an attacker.

We consider two possible approaches for path selection. In the first approach, the schedule is static and selecting one of the alternative paths is done entirely based on the position in the slot table at the moment of sending. The second option consists of dynamically selecting one of the paths at run-time based for example on a hardware random number generator. This non-deterministic behavior may be valuable from the security point of view, but it has a disadvantage in that several paths need to be reserved at the same time for the given channel.

A. Deterministic routing

For the deterministic routing approach, two hardware implementations are possible, as previously presented in [5]. One network architecture is based on distributed routing, while the other is based on source routing. In both cases, a global schedule computed at design time determines the movement of flits through the network. The schedule repeats periodically.

In the distributed routing network architecture 3, all network elements possess a fragment of the global schedule. The NI kernels contain information about when they are allowed to insert packets into the network, while routers contain a list of output ports each input should be forwarded to at each moment in time. That is to say, the route followed by a packet is determined only by its time of insertion into the network. This implementation supports multipath routing without any modifications.

In the source routing network architecture, both the schedules and the routes are stored in the network interfaces. A routing header is attached to each packet and will determine its path through the network. The necessary modifications consist of additional entries in the paths table and a path selection entry in the slot table (Figure 4).

B. Dynamic routing

The dynamic routing approach is only possible when using the source routing architecture previously presented. This is because the functionality of autonomously deciding between different destinations at runtime is not implemented inside routers.

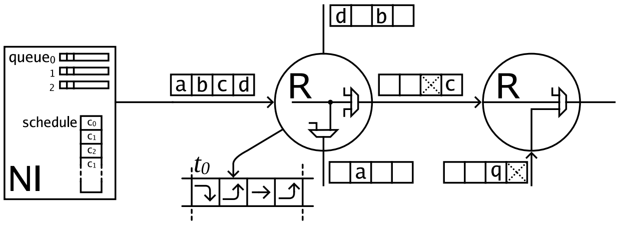

Fig. 3. Distributed routing architecture.

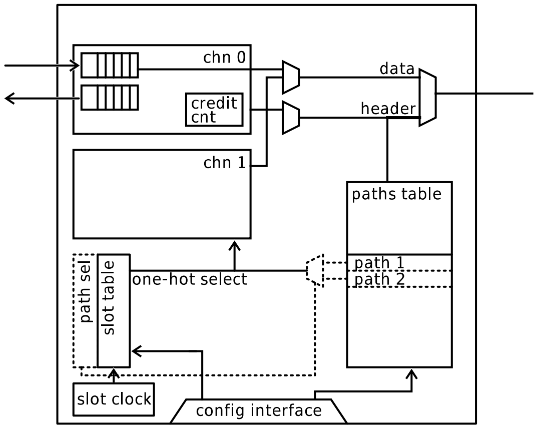

Fig. 4. Network interface, modifications for supporting multipath routing are represented with dotted line.

In the source routing architecture only one change is necessary compared to the deterministic routing approach, in that the path selection (“path sel”) signal in Figure 4 has to be generated by a random number generator instead of being obtained by table look-up.

True Random Number Generators [6] can be used for path selection, which protects against attacks based on multiple runs, although TRNGs have been show to be susceptible to attacks themselves[7]. Other approaches could consist of choosing the path based on the values that are being transmitted.

III. Slot allocation algorithm

In a previous study [5] we have shown that routing over several paths can provide a gain in terms of bandwidth, however, when using multipath for NoC security, the problem becomes more difficult for the following reasons:

1) it is now required to route certain channels over multiple paths rather than optional
2) the paths need to be disjoint, otherwise interception would be possible at a single point.

As previously mentioned, between the NI and its local router a single communication path exists and as a result that link has to be shared by all paths. In practice that may not represent a problem since the connection between the NI and router is

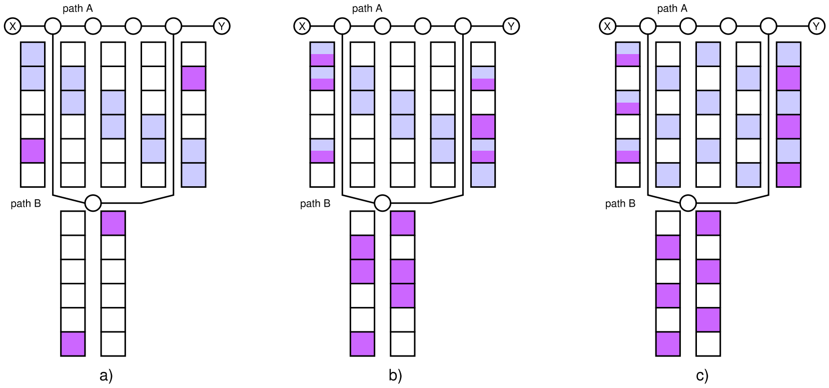

Fig. 5. Allocated slots, a) deterministic, b) non-deterministic grouped, c) non-deterministic sparse

chosen as a short back-to-back connection, or if necessary, the NI and the router can be integrated into the same component.

Pathfinding algorithm

We employ a backtracking algorithm to explore all pairs of disjoint paths up to a bounded length. The algorithm constructs solutions by alternatively adding segments to each of the two paths, keeping track of which slots are filled along each path and which are still usable. It is not mandatory to use the technique for all connections. The same algorithm may be used for performing the single-path search for the "insecure" connections, except we no longer attempt to allocate a second path.

Ideally both paths should arrive at the destination after the same number of hops, but that is not always possible. If one of the paths has already reached destination, we continue the search only for the second path. Once both paths reach the destination, we attempt to allocate a subset of the available slots on each path so that the bandwidth constraints are met.

For the dynamic (non-deterministic) modes, each of the paths must be able to support the entire channel bandwidth, while in the deterministic mode the bandwidth can be divided arbitrarily between the two channels with the restriction that the bandwidth on each channel is greater than 0. As it can be seen in figure 5, the non-deterministic mode requires the reservation of more network resources. When one connection uses consecutive slots, we have an additional choice of allowing or not allowing a path switch within the group of consecutive slots (we call this the grouped and the sparse mode). If the sparse mode is employed and there is a difference in length between the different paths we are forced to provide sufficient spacing between the allocated slots as illustrated in Figure 5c, If we did not provide this spacing, a flit sent on path A in slot 1 (Figure 5b) would arrive at the same time with a flit sent on path B in slot 2, thus creating a collision, which the Æthereal network does not allow.

We have tested our algorithm network configurations using the mesh topology and found it to be able to allocate multi-path connections in benchmarks consisting of a random connection set. Note that the mesh topology allows finding disjoint paths between any pair of nodes and the technique is only applicable on topologies that have this property.

IV. Conclusions and future work

In this paper we have proposed a method for protecting a NoC based interconnect against physical probing by using multi-path routing. Two configurations are proposed, one using deterministic and one using non-deterministic routing. The non-deterministic behavior provides protection against attacks consisting of multiple runs but has a higher penalty in terms of network resource usage.

Acknowledgements

This work was supported by the European Commission in the context of the Tera-Scale Multi-core Processor ARchitecture (TSAR) project #2A718.

References

[1] K. Tiri and I. Verbauwhede, "A digital design flow for secure integrated circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 7, pp. 1197–1208, July 2006.

[2] ——, "A vlsi design flow for secure side-channel attack resistant ics," in *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 58–63.

[3] S. P. Skorobogatov, "Semi-invasive attacks - a new approach to hardware security analysis," University of Cambridge, Technical Report UCAM-CL-TR-630, 2005.

[4] K. Goossens, J. Dielissen, and A. Radulescu, "Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Design & Test of Computers*, vol. 22, no. 5, 2005.

[5] R. Stefan and K. Goossens, "Multipath routing in tdm nocs," in *Proceedings of VLSI-SoC*, October 2009.

[6] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 109–119, 2007.

[7] S. Markettos, A.and Moore, *The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators*, 2009.