Packet Security with Path Sensitization for NoCs

Travis Boraten and Avinash Karanth Kodi
Department of Electrical Engineering and Computer Science
Ohio University, Athens, Ohio 45701
Email: tb286706@ohio.com, kodi@ohio.edu

***Abstract*—Hardware security is becoming a major concern as integrated circuits (IC) are exponentially growing thanks to technology scaling. With ICs reaching upwards of billions of transistors, detecting hardware trojans (HT) is like finding a needle in a haystack. Therefore, it becomes imperative to protect critical computing infrastructure from malicious attackers attempting to unearth vital information. Security enhancements should offer resiliency to limit their impact on overall chip performance as HTs are likely to slip through detection mechanisms. In this paper, we propose *packet-security (P-Sec)* a packet validation technique to protect compromised network-on-chip (NoC) architectures from fault injection side channel attacks and covert HT communication by merging two robust error detection schemes, namely algebraic manipulation detection (AMD) and cyclic redundancy check (CRC) codes. With P-Sec, applications containing sensitive and encrypted data can be protected from an ideal attacker using AMD codes at the cost of marginal area and power overhead in the network interface but with enhanced security on demand.**

I. Introduction

All too often, hardware security of integrated circuits (IC) in critical systems is taken for granted by architects and system designers. A malicious human attacker can then use micro-architectural vulnerabilities to alter functionality of a circuit at runtime. As aggressive transistor scaling continues, the ability to detect any malicious alterations which is achieved by additional transistors/circuits will become more difficult with each technology generation. Moreover, with the increasing globalization of corporations and the growing complexity of IC fabrication, it will be a significant challenge to protect intellectual property (IP) from falling into the hands of rogue fabrication plants. As future heterogeneous Multiprocessor System-on-Chip (MPSoC) architectures become prevalent and hundreds to thousands of cores are integrated[1], ensuring that third party IP cores are not tampered will prove to be a serious challenge. Making things more difficult is the diversity with which HTs can be implemented along with the payloads [2]. For example, they can be triggered sequentially or through combinational logic, externally or as ticking time bomb. When triggered, a HT may attempt to inject faults, corrupt data, steal sensitive data, cause denial of service, or render components inoperable via a kill switch.

As MPSoCs require scalable packet based Network-on-Chip (NoC) architectures to meet communication requirements, NoCs should provide security measures to protect sensitive data from fault injection attacks between IP cores while ensuring functional correctness to prevent systems from failing. As NoCs are expected to be the backbone of future on-chip communication, extensive research has expanded the field into several directions (power, fault tolerance, architecture) but security has gained exposure recently [3].

In this paper, we propose **Packet-Security (P-Sec)**, a highly confident packet validation scheme that takes a three-fold approach to enhance security in light of a compromised NoC by ensuring communication integrity through fault tolerance, functional correctness, and quality of service for critical applications. First, we propose a configurable fault tolerant router micro-architecture that can choose different levels of error detection encoding schemes. Second, by encoding packet header with path information, we enhance packet security within the router. Third, by prioritizing packet allocation, we provide quality of service and path sensitization. Our analysis shows that our proposed P-Sec can simultaneously provide both security and fault tolerance and is capable of thwarting attacks ranging from fault injection side channels to hardware trojans covertly stealing sensitive data - all of this while incurring less than 1% performance penalty with marginal power and area overhead. Specifically, this paper makes the following contributions:

[1] We propose a configurable fault tolerant router that can provide different levels of encoding with the goals of balancing power consumption with fault tolerance and security.

[2] Finally, we prioritize packet allocation to enhance quality of service and provide path sensitization for secure packets.

II. Related work

As HTs slip past chip validation and detection mechanisms, architects should design for security in future architectures and provide runtime solutions. Recent research in security for NoCs has shown to have the potential to improve system performance as well as system security. In *SurfNoC*[4], denial of service (DoS) and bandwidth depletion attacks are mitigated by partitioning network traffic in non-interfering multiplexing domains that reduces latency over traditional time division multiplexing. In [5], traditional light-weight switch-to-switch (s2s) error correction and link reshuffling are examined to isolate and mitigate HT-controlled links between routers. The link HT model used in [5] assumes combinational logic triggers that are tied to the links themselves, and only a few victim wires exists. If a link has more than two victim wires, error correction (hamming) will fail and injected faults will result in data corruption. In *Fort-NoCs*[6] a comprehensive approach was taken to eliminate the threat of a rogue router snooping on data and injecting its own packets into the network using data scrambling, packet certification, and node obfuscation. However, when faced with an ideal attacker, *Fort-NoCs* exhibits weakness in the encryption and authentication method as discussed later.

III. P-Sec: Packet Security

A. Attacker model

In Figure 1, we illustrate an ideal malicious attacker attempting to inject faults and break the encoding of packets.

978-3-9815370-6-2/DATE16/

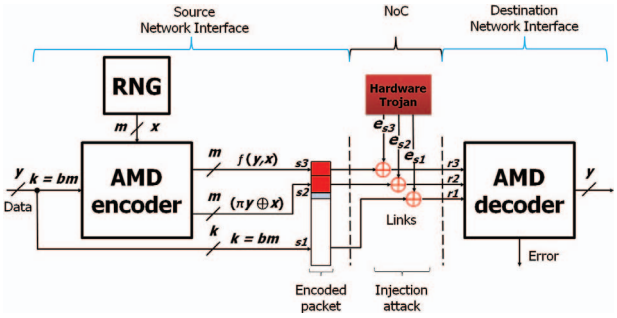

Fig. 1. The vulnerable links of compromised NoC protected from an ideal attacker by a (k,m,r) AMD encoder in the proposed P-Sec encoding scheme.

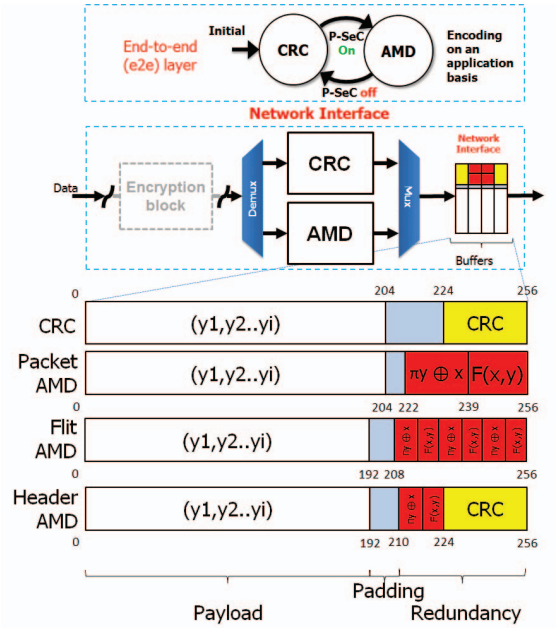

Fig. 2. The packet structure and end-to-end encoding states for CRC-32 and each (*k,m,r*) AMD encoding mode we evaluate in the network interface of each core. Encoding can be decided on-demand and on per application basis.

In this model, the attacker has control of inserting raw input y into the AMD encoder and thereby generating the error vectors e_{s1},e_{s2}, and e_{s3} on the target link. The HT triggering method for this model is negligible, as well is the method for controlling input y. With knowledge of different combination of inputs, the attacker can select error vectors $e1$,$e2$, and $e3$, in an attempt to eventually determine the sequence of errors required to mask codewords into another valid codeword. As encrypted communication is transmitted in MPSoC, an attacker may use this technique over time to obtain enough knowledge to decipher an encryption key by observing how the encoders and decoders react to a side channel attack. Such a side channel attack could be power [7], timing [8] or fault injection attacks [9]. In this paper, we limit our evaluation to fault injection due to HTs where the point of attack takes place within the network as opposed to other side channel attacks where the focus is on the encoding process itself.

B. Packet confidence with algebraic manipulation and detection codes

In our design we propose the use of algebraic manipulation detection (AMD) codes [10], [11] to boost protection of applications transmitting sensitive data between cores. AMD codes were originally proposed in [10], and has been evaluated in memory structures [11], but to our knowledge have not been studied to protect vulnerable links in NoCs. In a scenario with an ideal attacker, traditional codes such as SECDED, JTEC-QED, and CRC do not provide the robustness required to withstand such an attack as their error detection capabilities are low. Including CRC because the probability of detection diminishes with high fault rates. Strong AMD codes by definition cannot be masked into another valid codeword [12] for any error vectors e_{s1}, e_{s2}, e_{s3}, which is a unique advantage that traditional error correction and detection codes do not have, making them vulnerable to fault injection attacks. The redundant bits in an AMD code are a function of $f(x, y)$ shown in Figure 1 where y is the input data, x is a random number of size m, and $y = bm$ where b is chosen to lengthen y to a specific bit width. In our implementation for packet AMD encoding m is 17 bits and y is 204 bits, therefore b is 12. This translates to the packet formation as shown in Figure 2. The encoding function $f(y, x)$ is computed as $f(y, x) = y_1 x \bigoplus y_2 x^2 \bigoplus ... y_i x^i \bigoplus x^{b+3}$ and $\pi y = y_1 \bigoplus y_2 \bigoplus ... y_i \bigoplus x$. The complete AMD codeword forms the following structure, $C = (y, \pi y, f(x, y))$. If b is even the degree of the last term in $f(y, x)$ is x^{b+2} instead of x^{b+3}.

For applications working with sensitive or encrypted data, AMD encoding should be used. For all other non-critical traffic, packets will be encoded with CRC-32 to maintain minimal fault tolerance. The encoding used is designated in the header of each packet, along with the source, destination, packet type, and signature. The packet header for all traffic is then separately encoded using an AMD (23,7,7) codeword. With the packet header always protected, we can ensure a packet is always decoded correctly at the destination and that it is in fact at the correct destination. Any malicious or random alteration to the destination of other fields of the header will be detected. To prevent a valid header from being used on a maliciously crafted packet, after decoding, the packet signature adds another layer of validation that ensures any duplication of the packet will be caught. In the performance evaluation section we will highlight the advantages and disadvantages of each, along with a third option to encoded flits instead of entire packets to minimize additional overhead.

Figure 2, shows the modules required in the network interface for P-Sec and the packet makeup for both encoding structures. We also show the state diagram for encoding modes in each network interface. Network interfaces in P-Sec by default will encode packets with CRC-32 to maintain minimal fault coverage. In normal operating environments (not under attack) CRC is well capable of detecting faults in packets since the rate of faults naturally occurring are low. For cores sending sensitive data over the network, P-Sec is turned ON and they switch from CRC to AMD mode to protect sensitive packets from fault injection attacks.

C. Case Studies

In the network diagram of Figure 3, we highlight three scenarios a compromised NoC may encounter. In the first scenario, a source and destination are transmitting data across compromised links. The links in this scenario could be compromised by a simple HTs aiming to corrupt data or side channel attack. Since this traffic is not considered critical and

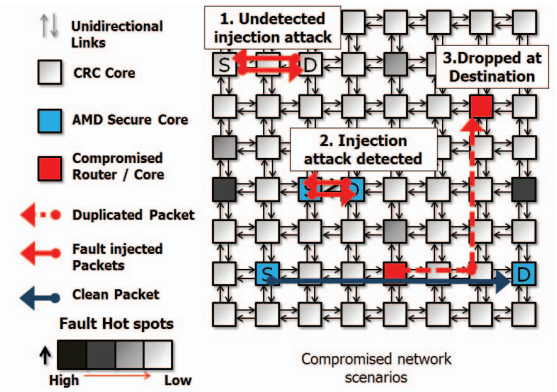

Fig. 3. In scenario 1, injected faults are undetected in insecure traffic. Scenario 2, fault injections are detected via AMD decoding for secure traffic. Scenario 3, a sniffed packet is duplicated and sent to a rogue core. The packet fails AMD header authentication and is dropped.

only CRC encoding is used, silent data corruption is likely as CRC may not detect any of the malicious alterations. However if AMD encoded traffic were to traverse a compromised channel as such in scenario two, the fault injection attack will not go unnoticed and silent data corruption will be avoided. Since P-Sec is primary concerned with end-to-end (e2e) encoding the location of the attack is still unknown. Depending on the duration and the strength of the attack, P-Sec may complement other mechanisms such as fault history logging and built-in self-test (BIST) in localizing the attack. If the attack can be localized, compromised components can be avoided and powered down. In the third scenario, we look at a situation visited in [6] where a compromised router is snooping on packets. When the compromised router finds its target packet, it duplicates the packet and sends it to another core with a rogue thread waiting to steal the data. In [6] they enable data scrambling, node obfuscation, and packet certification measures to circumvent this threat. With P-Sec, if the packet header is included in the AMD encoding, this threat will be mitigated upon decoding and the packets can be dropped within the network interface. A nack will be sent to the original source, since the source did not attempt to send data no invalid retransmission will be sent.

D. Security through Quality of Service

In P-Sec our primary concern is that packet validity and integrity is kept intact and that compromises do not breach beyond the network interface. We also want to ensure that packets are allowed to traverse the NoC properly without being blocked and enhance security furthermore through quality of service by giving applications higher priority through the micro-architecture pipeline. The packet header fields for our proposed design include an AMD encoded flag used to signal our prioritized arbitration unit that grants requests at random with higher weights for VCs allocated for applications needing higher security. When secure traffic does not exist, our prioritized arbitration unit acts as a traditional random arbitration block.

IV. Performance Evaluation

In order to compare and assess the effectiveness of AMD encoding, we compare several commonly used ECCs and evaluate the fault tolerant and security-advantages of P-Sec. We assume only one attack point in the network but the

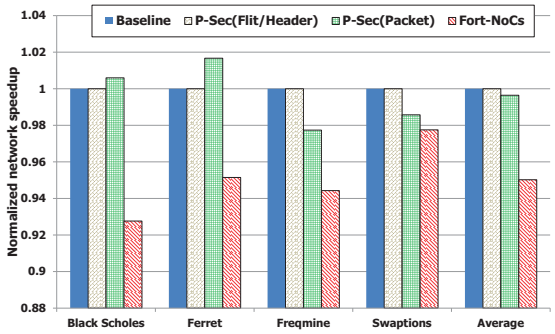

Fig. 4. Normalized network speedup of each packet authentication scheme, showing the performance overhead incurred for each application benchmark.

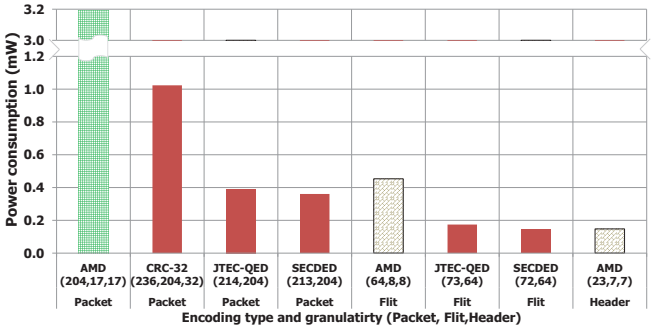

Fig. 5. The power consumption for each encoding method.

number of compromised links is orthogonal. Power and area were synthesized and optimized using the Synopsys Design Compiler tool using the TSMC 40 nm technology libraries with a 1.0 V supply voltage and 2 GHz operating frequency.

A. Network performance

To assess the performance impact of our proposed packet security and authentication scheme, we evaluated P-Sec against Fort-NoCs [6] on a typical 4 × 4, 64-core concentrated mesh topology using a in-house NoCs simulator. For closed-loop measurement, the full execution-driven simulator SIMICS from Wind River with the memory package GEMS was used to extract traffic traces from real applications. We assume a 2 cycle delay to access the L1 cache, a 4 cycle delay for the L2 cache, and a 160 cycle delay to access main memory. Figure 4 shows the normalized network speedup of P-Sec when compared to FortNoCs where the baseline network has no authentication. Results indicate P-Sec(Packet) reduces overhead compared to Fort-NoCs and incurs on average less than 1% performance penalty while improving fault tolerance and security coverage with our proposed encoding scheme. While Figure 4 indicates P-Sec(packet) improved speedup in two application benchmarks, this is misleading because flits encoded in P-Sec incur a two cycle AMD encoding penalty. The penalty however, is small relative to overall packet latency and in some cases, it aids in the relief of network congestion. Moreover if P-sec(Flit/Header) encoding is used, no performance overhead exists and speedup is unchanged because encoding fits within the network cycle.

B. Area, power and timing overhead

In Figure 5, 6 and Table I we display the power consumption, area, timing and information overhead required to

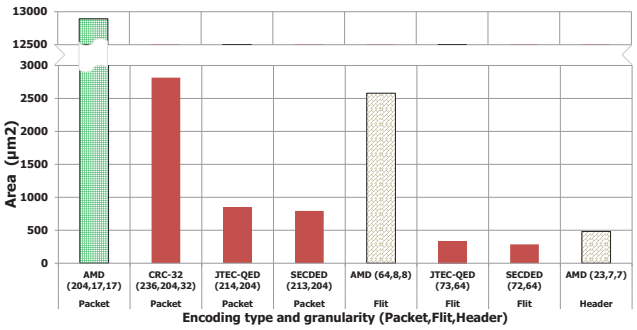

Fig. 6. The area overhead for each encoding method.

TABLE I. TIMING AND INFORMATION OVERHEAD.

Level	Encoder	Timing (ns)	Redundancy (bits)	Detection probability
Packet	CRC-32 (236,204,32)	1.09	32	All 1-3 bit, 32-bit burst, odd-bit errors
	SECDED (213,204)	0.33	9	All 2-bit
	JTEC-QED (214,204)	0.37	10	All 4-bit
	AMD (204,17,17)	**3.83**	**34**	$1-(12+2)2^{-17}$
Flit	SECDED (72,64)	0.24	8	All 2-bit
	JTEC-QED (73,64)	0.28	9	All 4-bit
	AMD (64,8,8)	**1.95**	**16**	$1-(8+2)2^{-8}$
Header	**AMD (23,7,7)**	**1.31**	**14**	$1-(3+1)2^{-7}$

encode packets in CRC-32, AMD, SECDED and JTEC-QED in packet, flit, and header granularity. AMD cost more power and area overhead for each level, however if the cost per total number faults detected is evaluated, AMD codes provide significant capability for the additional overhead. The additional power needed for P-Sec can be minimized as encoding can be chosen on demand per application.

Since AMD packet encoding has unfavorable power and performance overhead, we evaluated the cost of encoding with higher granularity for separate modes of operation. We found the cost of AMD encoding at the flit level not only reduced the power overhead from 10x to nearly 3x, but area overhead decreased and timing also fits within a 2 Ghz clock. Therefore AMD encoding within and for router switch-to-switch communication is possible, and the level of security had with packet encoding can now be sustained with flit encoding. The reduction in overhead is mainly due to cutback in multiplications required to compute $f(y,x)$, which was optimized by balancing the $y = bm$ equation for bit lengths and detection probability. Furthermore, with AMD header encoding we offer an additional mode that is used to sign and authenticate all traffic with similar overhead to flit SECDED.

C. Security

In this section, we discuss the security advantages AMD codes have over traditionally used ECCs in a compromised NoCs. In P-Sec, sensitive traffic is encoded with AMD. By using AMD, any malicious alteration of the data during traversal of the NoC will be detected at the destination core. In Table I, we list the error correction and detection capabilities of the compared codes. The superior error detection ability of AMD codes provide higher protection to NoCs from silent data corruption and side channel attacks. Further, AMD codes also compliment packet authentication techniques because the packet certificate can be included in the encoding. If the certificate is altered, such that packets are forwarded to a rogue core, when the packet reaches the network interface, decoding will compare the AMD signature, then detect the alteration and drop the packet.

V. CONCLUSIONS

In this paper we proposed P-Sec, a configurable packet validation and authentication scheme that takes a three-fold approach to enhance packet security. First, through our configurable fault tolerant router micro-architecture to adjust encoding with different levels of encoding for different levels of security. Second, by encoding packet header information to enhance security within the router, and third, by providing quality of service with secure packet prioritization and path sensitization. With our implementation of algebraic manipulation and detection codes (AMD) to protect packets, flits, and header information, HTs in links, routers, and side channel attacks can be thwarted in a compromised NoC. Performance results indicate P-Sec reduces overhead compared to Fort-NoCs and cost on-average less than 1% over a network with no packet authentication.

ACKNOWLEDGEMENT

This research was partially supported by NSF grants CCF-1054339 (CAREER), CCF-1318981, ECCS-1342657, CCF-1420718 and CCF-1541768.

REFERENCES

[1] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference*, ser. DAC '07. New York, NY, USA: ACM, 2007, pp. 746–749.

[2] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *Design Test of Computers, IEEE*, vol. 27, no. 1, pp. 10–25, Jan 2010.

[3] S. Sethumadhavan, A. Waksman, M. Suozzo, Y. Huang, and J. Eum, "Trustworthy hardware from untrusted components," *Commun. ACM*, vol. 58, no. 9, pp. 60–71, Aug. 2015.

[4] H. M. G. Wassel, Y. Gao, J. K. Oberg, T. Huffmire, R. Kastner, F. T. Chong, and T. Sherwood, "Surfnoc: A low latency and provably non-interfering approach to secure networks-on-chip," *SIGARCH Comput. Archit. News*, vol. 41, no. 3, pp. 583–594, Jun. 2013.

[5] Q. Yu and J. Frey, "Exploiting error control approaches for hardware trojans on network-on-chip links," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on*, Oct 2013.

[6] D. M. Ancajas, K. Chakraborty, and S. Roy, "Fort-nocs: Mitigating the threat of a compromised noc," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, ser. DAC '14, 2014.

[7] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '99, 1999, pp. 388–397.

[8] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '96, 1996, pp. 104–113.

[9] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, Nov 2012.

[10] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs, "Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors," in *Proceedings of the Theory and Applications of Cryptographic Techniques 27th Annual International Conference on Advances in Cryptology*, ser. EUROCRYPT'08, 2008, pp. 471–488.

[11] Z. Wang and M. Karpovsky, "Reliable and secure memories based on algebraic manipulation correction codes," in *On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International*, June 2012.

[12] Z. Wang and M. Karpovsky, "New error detecting codes for the design of hardware resistant to strong fault injection attacks," in *International Conference on Security and Management*, 2012.