Dynamic XY Routing for Security in NoCs

***Abstract*—Network on Chip is the technology used for interconnecting various components like routers, memory units etc,.. on the SoC systems. As it includes variety of internal components, security is one of the major threats. Any malicious routers present along the path to the destination could identify the destination address thereby increasing threat to the system. Anonymous routing solves this issue by encrypting the source and destination addresses along with the data sent. But the encryption techniques proposed in the previous systems like onion routing was resource consuming which produces an overhead. Dynamic XY routing algorithm eliminates this encryption, decryption overhead by reducing the number of decryptions to be done by the intermediary routers and at the maintaining security.**

Keywords—Network on Chip, Anonymous routing, XY routing, Dynamic XY routing, encryption.

I. Introduction

Modern IC fabrication involves outsourcing of various components of the chip rather than manufacturing under a same roof as an attempt to reduce time and cost of manufacturing. Since third party manufacturers are involved in various stages of IC fabrication, there is large possibility for introduction of threats in the form of hardware trojans [1] within the chip. The hardware trojans are designed with unique trigger patterns such that they are activated only on receiving some rare pattern of input. These varying structural and functional properties makes it impossible to develop a stable logic for detection of trojans. So, most of the hardware trojans are not detected during the testing phase of IC. The hardware trojans are generally introduced as malicious routers inside the chip. As NoC is the communication technique used in most SOC systems, the malicious routers are also included in the communication. This is a major threat as the security of the system including the software depends on the underlying hardware. Thus, the security threat is a major concern in NoC based SOC systems. Encryption of data is one of the traditional security measures followed in SoC systems. But some part of header information needs to be left unencrypted for the purpose of routing. The malicious routers with access to such unencrypted header information can still cause threats like misrouting. This problem has been overcome in anonymous routing algorithm.

II. Anonymous Routing

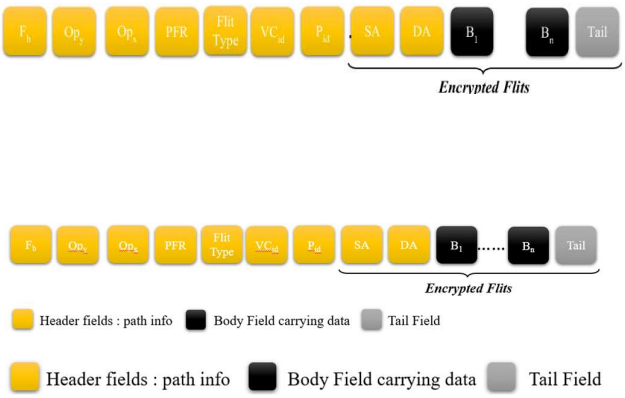
Anonymous routing is a routing technique where the source and destination addresses are not known by the intermediary routers i.e. sender and receiver remains anonymous. The intermediary routers know only the preceding and succeeding nodes and not the actual source and destination. Onion routing is one of the anonymous techniques where all the data including the header information are encrypted. The encryption and decryption are repeatedly done in all the intermediary routers as layers like that of an onion. NoCs are resource limited so this decryption performed in intermediary routers become an overhead. This has been resolved in [], where encryption is done only at the source router. Instead of using the destination address, strides are embedded in the header. The path to the destination address is chosen based on these strides.

III. Threat Model

The routers in chips forward both local and global data. So the third party groups aim to embed hardware trojans inside the routers. Confidentiality, integrity and availability are the major aspects of security of any system. Compromising atleast of one these three factors is the main goal of introducing hardware trojans inside the routers by stealing or spoofing the data. When the routing information in the header like source and destination address are left as plain text, this could be utilized by the malicious routers to extract the data and pass it to the adversary for further threat generation. Eavesdropping, stealing and misrouting are the main threats that are to be addressed in this algorithm. Data corruption is not the scope of this paper.

IV. Proposed Anonymous Routing Technique

The route chosen in previous systems is not adaptive. There is still a chance that the malicious router can get few information on the destination. The algorithm proposed in this paper meets this requirement by choosing adaptive routes. This anonymous routing technique is a subset of dynamic XY routing algorithm. Dynamic routing algorithm is an enhanced version of XY routing algorithm, where the packet is allowed to choose one direction dynamically at each node. Those two directions are initially chosen by computing the difference between x and y coordinates i.e if x difference is positive, then east direct is taken or else west direction is chosen. Similarly, either north or south is chosen based on difference in y.



A. Packet format

In order to encrypt the header data before transfer, some supplementary data are computed in the source router and embedded in the header segment of the packet. Flip Bit (F_b) gives the direction by which the flip should occur. If the flip bit is 0, then the flip should occur in x direction or if it is 1,

then it should flip in y direction. Output Port on x-axis (O_{px}) gives the direction in which the packet should travel along x-axis. 0 indicates East direction and 1 indicates South. Output Port on y-axis (O_{py}) gives the direction in which the packet should travel along y-axis. 0 indicates North and 1 indicates South direction. Pre Flip range (PFR) gives the total no. of hops required to be taken before flip occurs.

Algorithm 1: Path computation

Data: $Packet(P_s, P_d)$
Result: Op_x, Op_y, pfr
$\triangle x \leftarrow P_d.x - P_s.x;$
$\triangle y \leftarrow P_d.y - P_s.y;$
$pfr \leftarrow abs(abs(\triangle x) - abs(\triangle y));$
if $\triangle x \geq 0$ **then**
  $Op_x \leftarrow 0;$
else
  $Op_x \leftarrow 1;$
end
if $\triangle y \geq 0$ **then**
  $Op_y \leftarrow 0;$
else
  $Op_y \leftarrow 1;$
end
if $abs(\triangle x) \geq abs(\triangle y)$ **then**
  $flipbit \leftarrow 0;$
else
  $flipbit \leftarrow 1;$
end
while $True$ **do**
  $outportdirn \leftarrow getoutportdirn();$
end
$return;$

B. Initial Computation

As a first step before transfer, the above discussed supplementary data are computed and embedded in the header of the packet. All other data including the source and destination address except these supplementary data are encrypted. south). The flipbit is initially set at Source based on the value of Δx and Δy. If Δx is greater, then flipbit is set to 0, if Δy greater then flipbit is set to 1. The absolute difference between Δx and Δy is the hop count and is embedded as PFR.

C. Path traversal

The pre flip range gives the minimum number of hops required in the direction of flip bit to reach the destination. Since PFR is finite, the packet traverses in flipbit direction for PFR number of times. Once PFR becomes zero, the packet takes zig zag path by alternatively taking direction of O_{px} and $O_{py,}$ infinitely until the destination router is reached.

Subroutine 2: getoutportdirection()

if $pfr \geq 1$ **then**
  if *(flipbit)* **then**
    $outportdirn \leftarrow Op_y;$
  else
    $outportdirn \leftarrow Op_x;$
  end
  $pfr \leftarrow pfr - 1$
else
  if *(flipbit)* **then**
    $outportdirn \leftarrow Op_y;$
  else
    $outportdirn \leftarrow Op_x;$
  end
  $flipbit \leftarrow \sim flipbit;$
end
$return\ \ outportdirn;$

REFERENCES

[1] S. Charles, Y. Lyu and P. Mishra, "Real-time Detection and Localization of DoS Attacks in NoC based SoCs," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019, pp. 1160-1165, doi: 10.23919/DATE.2019.8715009.

[2] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. 2014. Hardware Trojan attacks: Threat analysis and countermeasures. Proc. IEEE 102, 8 (2014), 1229–1247.