Segurança em Redes-em-Chip: Conceitos e Revisão do Estado da Arte

Sidnei Baron[1]
Michelle Silva Wangham[1]
Cesar Albenes Zeferino[1]

Resumo: As Redes-em-Chip buscam os requisitos de escalabilidade de comunicação em sistemas computacionais integrados em uma única pastilha de silício. Assim como sistemas distribuídos tradicionais, um sistema integrado e sua rede são suscetíveis a ataques às suas propriedades de segurança. Este artigo apresenta um levantamento bibliográfico realizado para caracterizar as técnicas que têm sido utilizadas para prover segurança em sistemas integrados baseados em Redes-em-Chip. Os trabalhos foram classificados quanto ao tipo de ataque, à propriedade de segurança afetada, ao mecanismo de segurança utilizado e ao componente adotado na implementação da solução proposta. O artigo identifica os principais ataques às propriedades de segurança que foram abordados por esses trabalhos, bem como as soluções mais utilizadas para melhorar a segurança das redes.

Palavras-chave: Sistemas Integrados, Redes-em-Chip, Segurança.

Abstract: Networks-on-Chip (NoCs) aim at meeting the communication scalability required by Systems-on-Chip (SoCs), which are computing systems integrated into a single chip. As well as traditional distributed systems, a SoC and its network are susceptible to attacks to their security properties. This paper presents a survey that was conducted in order to identify the techniques that have been applied to provide security in NoC-based SoCs. The analyzed works were classified regarding the addressed attacks, the affected security properties, the adopted security mechanisms, and the components used to implement the proposed solutions. Concluding, the paper identifies the main attacks addressed by these works and the solutions most used to improve the network security.

Keywords: Systems-on-Chip, Networks-on-Chip, Security.

[1] Laboratório de Sistemas Embarcados e Distribuídos, Universidade do Vale do Itajaí
Rua Uruguai, 458 – Centro – CEP: 88302-202 – Itajaí, SC, Brasil
{sbaron, wangham, zeferino} @univali.br

1 Introdução

O advento dos processos submicrônicos viabilizou a integração de sistemas computacionais completos em uma única pastilha de silício. Esses sistemas, denominados sistemas integrados ou SoCs (*System-on-Chip*), baseiam-se no reuso de blocos de *hardware* previamente projetados e verificados, os quais são chamados de blocos de IP (do inglês, *Intellectual Property blocks*) ou núcleos (do inglês, *cores*) [1].

A busca pela redução do consumo de energia e pelo aumento no desempenho dos sistemas computacionais impulsionou o desenvolvimento de SoCs com múltiplos núcleos (*multi-core*), permitindo que um único *chip* possua de centenas a milhares de núcleos integrados [2].

Com o aumento do número de núcleos em um único *chip*, as necessidades de escalabilidade de largura de banda de comunicação se tornaram criticamente importantes. Para suprir essas necessidades, as redes chaveadas estão substituindo os barramentos compartilhados e os canais ponto a ponto na comunicação entre os núcleos de um sistema integrado [2]. Essas redes são denominadas *Networks-on-Chip* (NoCs), sendo que, em português, são adotados os termos Redes-em-Chip e redes intrachip.

Os primeiros a propor o uso de redes chaveadas na comunicação entre componentes de um sistema computacional integrado em um único *chip* foram Tewksbury, Uppuluri e Hornak [3], em 1992. A primeira solução experimental descrita na literatura foi a SPIN (*Scalable Programmable Interconnection Network*), apresentada por Guerrier e Greiner [4] em 1999. Já o termo *Network-on-Chip* (NoC) foi proposto por Hemani et al. [5] no ano 2000, tendo sido amplamente adotado pela comunidade após a publicação do artigo de Benini e De Micheli [6] na revista *Computer*, em 2002. A primeira NoC desenvolvida no Brasil foi a rede SoCIN (*SoC Interconnection Network*), a qual foi apresentada por Zeferino e Susin em 2003 [7].

A preocupação com a segurança da computação é um fator significativo no desenvolvimento e na aplicação dos sistemas computacionais em toda a sociedade. Como todos os sistemas computacionais, um SoC (seja este baseado em barramentos compartilhados, canais ponto a ponto dedicados ou redes chaveadas) também é alvo de ataques à segurança. Por exemplo, os invasores podem tentar extrair alguma informação do sistema com o objetivo de obter dados pessoais dos seus usuários ou no intuito de burlar a licença de uso de algum *software* executado pelo sistema. Outro ataque pode ser aquele que tem por objetivo a degradação do desempenho do sistema por meio de ações que levem à negação de serviços dentro do SoC.

Em um sistema integrado que utilize uma NoC como infraestrutura de interconexão, a rede é o coração do sistema, pois gerencia todas as comunicações e, por isso, os ataques contra a NoC são críticos [8]. É possível bloquear ataques de uma tarefa (atacante) a outra (vítima) por meio da implementação de mecanismos de segurança nos componentes da rede [9].

A segurança em NoCs tem sido um dos alvos de estudo mais recentes dos grupos de pesquisa. Isso se justifica porque a complexa comunicação das NoCs pode trazer novas deficiências ao sistema, o que pode ser crítico e deve ser cuidadosamente estudado e avaliado [10]. Por outro lado, as NoCs podem contribuir para a segurança do sistema, fornecendo um meio ideal para monitorar o comportamento do SoC e detectar ataques específicos.

Dentro desse contexto, este artigo apresenta um estudo realizado para identificar quais técnicas têm sido utilizadas para prover segurança em SoCs baseados em NoC. Os trabalhos selecionados e analisados foram classificados quanto ao tipo de ataque, a propriedade de segurança afetada, o mecanismo de segurança utilizado e o componente adotado para implementar a solução proposta.

Este artigo está organizado em seis seções. Nesta seção, foi apresentado o contexto do estudo realizado. A Seção 2 apresenta uma breve revisão de conceitos sobre NoCs, enquanto a Seção 3 sumariza os conceitos base de segurança em sistemas computacionais. Na Seção 4, são discutidos os primeiros estudos publicados na literatura e o estado da arte sobre segurança em NoCs. Na Seção 5, é apresentada uma análise comparativa desses trabalhos, identificando os principais alvos de investigação e suas soluções preferenciais. Concluindo, a Seção 6 apresenta as considerações finais do artigo.

2 Redes-em-Chip

Uma NoC é composta basicamente de três componentes: interface de rede, enlace e roteador [11]. A interface de rede, ou NI (*Network Interface*), implementa a lógica que realiza a adaptação entre os protocolos de comunicação dos núcleos e da NoC. A NI deve ser projetada de modo a facilitar o seu reuso com núcleos de diferentes plataformas e também de diferentes arquiteturas de comunicação.

Um enlace liga dois pontos na rede, sendo que esses pontos podem ser um roteador ou um núcleo. O enlace pode possuir um ou dois canais físicos de comunicação. Os mais utilizados são os enlaces *full-duplex*, os quais são constituídos de dois canais unidirecionais opostos de modo a permitir a transferência simultânea de informação nas duas direções do enlace [12].

Um roteador é composto de uma estrutura de chaveamento (denominada *crossbar*) e uma lógica de controle para roteamento e arbitragem, além de portas para comunicação com outros roteadores e/ou com os núcleos [12]. Essas portas possuem controladores de enlace que regulam o tráfego das informações que entram e saem do roteador.

Uma NoC pode ser descrita pela sua arquitetura e pelos mecanismos de comunicação por ela utilizados. A topologia define a estrutura da rede, enquanto os mecanismos de comunicação definem a forma pela qual ocorre a transferência das mensagens pela rede. Os mecanismos de comunicação incluem: controle de fluxo, chaveamento, memorização, roteamento e arbitragem [12].

A topologia de uma NoC determina a sua organização física e as conexões entre os núcleos e os canais na rede. A topologia também define a quantidade de saltos (roteadores) que a mensagem deve atravessar para chegar ao seu destino [2].

O controle de fluxo é o mecanismo de comunicação que gerencia a alocação dos *buffers* e dos enlaces da rede. Um bom protocolo de controle de fluxo diminui a latência e não sobrecarrega os recursos da rede, melhorando o seu desempenho por meio do compartilhamento efetivo dos *buffers* e dos enlaces [2].

O mecanismo de chaveamento define a forma pela qual os dados são transferidos de um canal de entrada de um roteador para um dos seus canais de saída [2]. Existem dois tipos básicos de técnica de chaveamento. O primeiro é o chaveamento por circuito, o qual se baseia no estabelecimento de um caminho completo entre o núcleo de origem e o destinatário da mensagem. O segundo é o chaveamento por pacotes, o qual consiste na divisão das mensagens em pacotes que reservam seus caminhos dinamicamente na medida em que avançam em direção ao destinatário. Existem ainda variantes do chaveamento por pacotes, incluindo as técnicas denominadas *Store-and-Forward* (SAF), *Virtual Cut-through* (VCT) e *Wormhole*.

A memorização (ou *buffering*) diz respeito ao mecanismo de armazenamento de pacotes bloqueados dentro do roteador [2]. Os *buffers* de memória podem ser posicionados nos canais de entrada do roteador, nos canais de saída e/ou podem ser centralizados e compartilhados por mais de um canal. Os *buffers* também podem ser utilizados na interface de rede [11].

O roteamento é o método usado para decidir o caminho que a mensagem irá tomar através da rede para chegar ao seu destinatário. O desempenho da rede depende fortemente do algoritmo de roteamento utilizado [2].

Enquanto o roteamento é um mecanismo de seleção de saída do pacote no roteador, a arbitragem determina qual canal de entrada pode utilizar um determinado canal de saída do roteador. A arbitragem é fundamental para a resolução de conflitos decorrentes da existência de múltiplos pacotes competindo por um mesmo canal de saída [2].

Uma mensagem é quebrada em pacotes e os pacotes, por sua vez, são divididos em unidades de transferência de tamanho fixo que são denominadas *flits* (*flow control units*). Um pacote é formado por *flits* de cabeçalho, corpo e cauda [2]. O corpo do pacote é também chamado de carga útil, enquanto a cauda pode ser denominada terminador. Em geral, um *flit* tem o tamanho de um a quatro *phits* (*physical units* – unidade física que corresponde à largura do canal) para conter, no mínimo, a informação necessária ao roteamento do pacote pelos roteadores, ou seja, o cabeçalho. A Figura 1 ilustra a diferença entre mensagem, pacote, *flit* e *phit*.

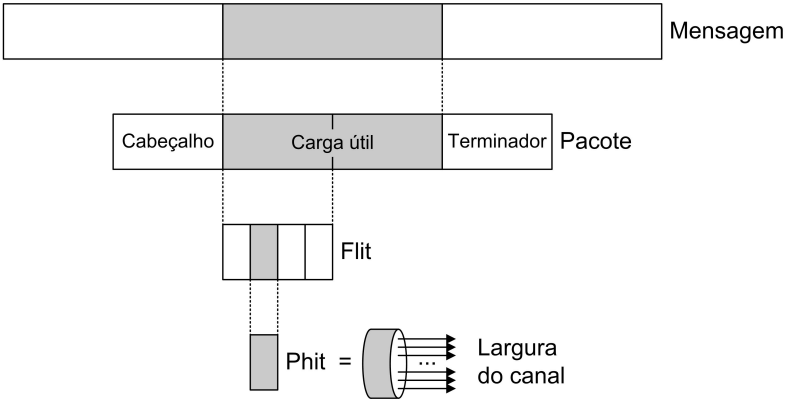

Figura 1. Mensagem, pacote, *flit* e *phit*

3 Segurança em Sistemas Computacionais

A preocupação com a segurança da computação é um fator relevante no desenvolvimento e na aplicação dos computadores em toda a sociedade [13]. O conceito de segurança da informação consiste na preservação das propriedades de segurança [14], sendo que a violação da segurança ocorre quando há a quebra de uma ou mais dessas propriedades [15]. Conforme discutido por Landwehr [13], cinco propriedades de segurança devem ser garantidas em um sistema seguro:

1. *Confidencialidade*: garantia de que a informação não será divulgada sem autorização;
2. *Integridade*: garantia de que a informação não será modificada sem autorização;
3. *Disponibilidade*: garantia de que a informação ou recurso estará acessível para os usuários autorizados sempre que requisitados;
4. *Autenticidade*: garantia de que cada entidade é quem afirma ser; e
5. *Não repúdio*: garantia de que a participação de uma entidade em uma dada transação ou evento não pode ser negada.

Quatro aspectos estão relacionados à segurança de um sistema computacional [16]:

- *Vulnerabilidade*: é uma fraqueza em um sistema quanto aos seus procedimentos de segurança, controles internos ou implementação que pode ser explorada por uma fonte de ameaça;
- *Ameaça*: é qualquer circunstância ou evento com potencial para afetar negativamente as operações de um sistema de informação por meio de um acesso não autorizado, da destruição, da divulgação, da modificação de informações e/ou da negação de serviço;
- *Risco*: é a medida da extensão em que uma entidade está ameaçada por uma potencial circunstância ou evento. Pode ser representado como uma função dos impactos adversos que surgiriam se a circunstância ou evento ocorresse, bem como da probabilidade de ocorrência;
- *Intrusão*: é o ato de contornar os mecanismos de segurança de um sistema. Ou seja, é quando um ataque é bem sucedido.

A arquitetura de segurança OSI (*Open System Interconnection*)[2], descrita em [17], oferece uma estrutura sistemática para definir os requisitos de segurança e caracterizar as técnicas para satisfazer esses requisitos. Essa arquitetura engloba serviços de segurança, ataques e mecanismos. Um serviço de segurança aumenta a segurança dos sistemas de processamento de dados e as transferências de informação. Os serviços de segurança servem para frustrar ataques ao sistema e utilizam um ou mais mecanismos para garantir as propriedades de segurança.

Um ataque à segurança pode ser definido como qualquer ação que comprometa a segurança da informação. Esses ataques são classificados em ataques passivos e ataques ativos. Um ataque passivo tenta monitorar o sistema com o objetivo de obter informações, mas não afeta os recursos ou altera os dados envolvidos. Por outro lado, os ataques ativos envolvem alguma modificação do fluxo de dados ou a formação de um fluxo falso, afetando a operação do sistema [17]. Os ataques passivos e ativos podem ser divididos nas categorias apresentadas no Quadro 1.

[2] A arquitetura de segurança OSI foi proposta pela ITU (*International Telecommunications Union*), mas não está relacionada com o modelo de rede em sete camadas, conhecido como modelo de referência OSI.

Quadro 1. Classificação dos ataques passivos e ativos

Ataque	**Categoria**	**Descrição**
Passivo	Liberação de conteúdo da mensagem	Ocorre quando uma informação é captada e o seu conteúdo é lido pelo atacante
	Análise de tráfego	Ocorre quando o tráfego da troca de uma informação (criptografada ou não) é analisado para identificar padrões nas mensagens
Ativo	Disfarce	Ocorre quando uma entidade finge ser outra entidade
	Repetição	Ocorre quando os dados são capturados passivamente e, subsequentemente, retransmitidos para produzir um efeito não autorizado
	Modificação da mensagem	Ocorre quando alguma parte da mensagem original é alterada para produzir um efeito não autorizado
	Negação de serviço	Ocorre quando há um impedimento ou inibição do uso ou gerenciamento normal das instalações de comunicação

O mecanismo de segurança é um processo projetado para detectar, impedir ou permitir a recuperação de um ataque à segurança [17]. Os mecanismos de segurança específicos podem ser categorizados como:

- *Cifragem*: mecanismo que usa algoritmos matemáticos para transformar dados em um formato que não seja prontamente decifrável;
- *Assinatura digital*: mecanismo que utiliza a transformação criptográfica (ou dados anexados) para permitir que o destinatário comprove a origem e a integridade dos dados;
- *Controle de acesso*: mecanismo que impõe direitos de acesso aos recursos;
- *Integridade dos dados*: mecanismo que busca garantir a integridade de dados ou fluxo de dados;
- *Troca de informações de autenticação*: mecanismo que busca garantir a identificação de uma entidade por meio da troca de informações; e
- *Preenchimento de tráfego*: mecanismo que realiza a inserção de bits nas lacunas de um fluxo de dados para frustrar as tentativas de análise de tráfego.

4 Soluções de Segurança para Redes-em-Chip

Esta seção apresenta uma descrição da evolução das pesquisas sobre o provimento de segurança em SoCs baseados em NoCs. Ela é fruto de uma revisão sistemática realizada com o objetivo de caracterizar as principais contribuições apresentadas na literatura e serviu de referencial para uma pesquisa sumarizada no final desta seção. Os trabalhos são descritos pela ordem temporal de publicação, partindo do primeiro trabalho sobre o tema, publicado em 2003.

Gebotys e Gebotys [18] propuseram o uso de um ambiente de segurança em NoC para evitar acesso não autorizado, invasão de *software*, cópia não autorizada do núcleo e ataques de energia e de eletromagnetismo[3]. No mecanismo proposto, foi desenvolvida uma camada na interface de rede para implementação de segurança. Os núcleos protegidos por uma interface de rede com essa camada são denominados *Secure Core*. Também foi implementado um núcleo centralizado para armazenar as chaves do sistema, o qual é denominado *Key-keeper*. Antes de ocorrer a comunicação entre os núcleos conectados à NoC, a interface de rede do *Secure Core* solicita ao *Key-keeper* a sua chave privada, a qual está armazenada de forma segura na memória. Essa comunicação entre o *Key-keeper* e a interface de rede é também criptografada, porém com uma chave de rede previamente armazenada no próprio núcleo. Após o armazenamento da chave privada na interface de rede, a comunicação é feita por meio de criptografia de chave pública e privada. A solução foi avaliada e apresentou um baixo impacto no desempenho do sistema (menor que 11%).

Diguet, Evain e Vaslin [19] propuseram uma solução para autenticação de entidades e de integridade de dados. O mecanismo proposto é composto de um gerenciador de segurança e de interfaces de rede seguras. As transações entre os núcleos são monitoradas e somente autorizadas se forem permitidas pela política de segurança. A comunicação é feita por meio de canais virtuais dedicados, separando os fluxos de dados da aplicação dos fluxos de dados de segurança. A solução foi implementada e verificada por simulação, com o uso da ferramenta μSpider [20], e por prototipação, usando a ferramenta ISE da Xilinx. Os resultados mostraram que o custo de área da solução proposta é 45% maior que o da implementação sem a solução de segurança. O sobrecusto é principalmente devido ao canal seguro implementado nos roteadores, que passaram a ter dois canais virtuais ao invés de um único canal, como no roteador não seguro.

Fiorin, Palermo e Silvano [21] propuseram algumas soluções para evitar ataques de negação de serviço (DoS – *Denial-of-Service*), extração de informação secreta e *hijacking* (acesso de escrita em área segura) em NoCs. A primeira solução proposta, denominada APU (*Address Protection Unit*), busca restringir o acesso à memória, evitando *buffer overflow* ou ataques que possam roubar ou modificar informações sensíveis da memória. A segunda

[3] Ataque que visa extrair informações através do vazamento de informações por campos eletromagnéticos.

solução busca prover garantia de banda por meio do uso de técnicas de qualidade de serviço (QoS – *Quality-of-Service*) para reservar uma certa quantia de banda para cada núcleo, evitando assim que códigos maliciosos ou mal escritos possam consumir toda a banda da rede. A terceira solução proposta consiste em um autômato de segurança que monitora as transações a fim de detectar ataques conhecidos de/para um núcleo na rede. A quarta solução consiste de um mecanismo de IDS (*Intrusion Detection System*) que monitora o canal de comunicação entre os núcleos para detectar violações de segurança, tais como: (*i*) ocupação de *buffers*; (*ii*) comportamento anômalo do gerenciamento de energia; (*iii*) acesso não autorizado a locais de memória seguros; e (*iv*) violações na execução de rotinas críticas. Nenhuma das soluções propostas chegou a ser implementada nesse trabalho.

Fiorin et al. [22] propuseram uma solução para proteção dos dados no acesso à memória em sistemas baseados em NoC. A solução consiste em uma arquitetura de rede segura baseada em uma unidade de proteção de dados (DPU – *Data Protector Unit*) integrada nas interfaces de rede da NoC. A DPU garante acesso seguro à memória e a periféricos mapeados em memória, habilitando o acesso ao espaço de memória somente se houver autorização. O acesso é filtrado considerando não somente o endereço da memória, mas, também, a operação requisitada (leitura, escrita e execução) e o estado do requisitante da operação (usuário/supervisor, modo seguro/inseguro). A configuração em tempo de execução da parte programável da DPU é gerenciada por uma unidade central denominada NSM (*Network Security Manager*). Duas alternativas foram apresentadas: DPU na origem, chamada de DPU@INI (DPU *at Initiator* NI), e DPU no destino, chamada de DPU@TNI (DPU *at Target* NI). A solução proposta foi implementada por prototipação usando o processador ARM920T e uma memória de 16Kb SRAM. Foram utilizadas as ferramentas Synopsys Design Compiler e Prime Power com a biblioteca de tecnologia 0.13µm HCMOS9GPHS da STMicroelectronics para analisar o consumo de energia, o sobrecusto de área e o atraso do caminho crítico para os dois tipos de implementação, DPU@INI e DPU@TNI. Os resultados mostraram que a implementação da DPU tem um sobrecusto de 17% de área e de 7,5% de energia, sem impactar significativamente no desempenho do sistema.

Fiorin, Palermo e Silvano [10] propuseram uma solução para evitar a extração de dados sensíveis e ataques de DoS. O mecanismo proposto consiste em um sistema de monitoração para detectar violações de segurança em uma NoC. Informações coletadas na interface dos núcleos são enviadas a uma unidade central (NSM – *Network Security Manager*) para neutralizar ações exercidas pelo atacante. A arquitetura possui dois tipos de sondas (*probes*): (*i*) *Illegal Access Probe* (IAP), que detecta tentativas não autorizadas de acesso a locais de memória; e (*ii*) DoS *Probe* (DoSP), que detecta um comportamento não natural de tráfego. A solução foi implementada por prototipação usando a tecnologia 0.13µm HCMOS9GPHS da STMicroelectronics. Os experimentos realizados no trabalho avaliaram os sobrecustos de área e de energia, comparando-os com os da solução sem o monitoramento de segurança. Os resultados mostraram que o sobrecusto da implementação da sonda IAP na NI não é significativo (0,02% em área) e o da sonda DoSP aumenta linearmente com a inclusão de configurações de entrada. O sobrecusto total é de 34,7% se comparado com a solução sem a monitoração de segurança.

Sepúlveda, Strum e Chau [9] propuseram uma solução de níveis de segurança na dimensão de QoS para controle de acesso, autenticação e disponibilidade dos dados. O mecanismo proposto define quatro níveis de segurança para cada serviço, do Nível 0 (sem segurança) ao Nível 3 (segurança máxima). A solução foi implementada tanto no roteador como na interface de rede, usando o protocolo de comunicação OCP/IP (*Open Core Protocol – Intellectual Property*). A implementação na interface de rede foi realizada no núcleo escravo. O propósito do serviço de controle de acesso é permitir que sejam feitas somente transações autorizadas, atuando como um *firewall*. A implementação de diferentes níveis de controle de acesso usa três mecanismos que verificam: (*i*) a origem do pacote; (*ii*) o tipo de transação; e (*iii*) a regra do mestre. O propósito do serviço de autenticação é assegurar-se quanto à identidade do emissor de um pacote recebido, verificando: (*i*) a origem do pacote; (*ii*) o cálculo do caminho; e (*iii*) o par mestre-escravo. O propósito do serviço de disponibilidade é garantir que um recurso da rede possa ser utilizado quando requerido. A implementação foi realizada da seguinte forma: (*i*) adicionando canais virtuais; (*ii*) modificando a arbitragem; e (*iii*) adotando um chaveamento híbrido (baseado em circuitos e em pacotes). Os experimentos de avaliação foram realizados por meio de um ambiente de simulação SystemC TLM (*Transaction-level Modeling*). Para efeito de comparação, foram implementados os três mecanismos, tanto no roteador quanto na interface de rede do núcleo escravo, usando uma NoC com topologia em malha 2D 4×4 e alterando os níveis de segurança de cada mecanismo. Os resultados mostraram que é mais eficiente implementar os serviços propostos no roteador do que na interface de rede. A implementação de controle de acesso e da autenticação aumentou o consumo de energia e a latência, sendo que somente no serviço de disponibilidade é que essas métricas são reduzidas devido ao uso do mecanismo de chaveamento híbrido.

Stefan e Goossens [23] propuseram uma solução contra ataques de análise de tráfego durante a comunicação entre núcleos em uma NoC. O mecanismo proposto baseia-se no uso de caminhos alternantes (determinístico) ou de caminhos aleatórios (não determinístico) para enviar os dados entre os núcleos, dificultando a extração de dados, caso um atacante consiga "escutar" um dos canais da rede. A solução foi implementada alterando a interface de rede da NoC Æthereal [24] e os resultados mostraram que a implementação do algoritmo não determinístico protege a NoC contra os ataques, mas com a penalidade da alta utilização dos recursos da rede.

Huffmire et al. [25] propuseram uma solução para a comunicação entre núcleos de um sistema embarcado por meio dos conceitos de fossos e de pontes levadiças. Os fossos separam os núcleos, enquanto as pontes levadiças permitem controlar a interação entre eles. A solução proposta foi implementada com prototipação em dispositivo lógico programável do tipo FPGA (*Field Programmable Gate Array*) e os experimentos de avaliação foram realizados usando o *benchmark* MCNC (*Microelectronics Center of North Carolina*). Os resultados mostraram que o sobrecusto de área dos fossos é mínimo, no pior caso é de 2,61% e no melhor caso é de 1,05%. Em fossos grandes, o período de relógio é melhor, porém o sobrecusto de área é maior. Embora a pesquisa originalmente não tenha sido realizada com foco em NoCs, os autores comentam que poderiam estender o conceito de pontes levadiças para suportar esse tipo de rede.

Lukovic e Christianos [26] propuseram uma solução de segurança em diferentes níveis de projeto para MPSoCs (*Multiprocessor* SoCs) baseados em NoC, monitorando a execução da aplicação e validando se o comportamento é correto. A solução pode integrar vários tipos de abordagem de segurança para proteção contra ataques específicos. O mecanismo proposto é constituído de uma estrutura hierárquica de agentes de segurança. Os quatro tipos de agentes são: (*i*) agente específico de ataque (ASA – *Attack Specific Agent*); (*ii*) agente de segurança local (LSA – *Local Security Agent*); (*iii*) agente de segurança do *cluster* (CLSA – *Cluster Security Agent*); e (*iv*) agente de segurança central (CSA – *Central Security Agent*). Para garantir uma comunicação segura entre esses agentes, os autores propuseram o uso de uma NoC adicional (denominada S-NoC ou *Secure* NoC) em paralelo à NoC regular já existente. Como estudo de caso, foi considerado um cenário de ataque de injeção de código realizado via *buffer overflow* com sobrescrita em posições da pilha (*e.g.* no endereço de retorno de um procedimento). Foi considerado também um sistema com um processador MicroBlaze conectado às redes regular e segura por meio de uma interface de rede melhorada. Essa interface possui um agente específico de ataque especializado em proteger a pilha, apresentado em [27], e um agente de segurança local. O estudo de caso foi implementado e verificado em FPGA (modelo Virtex-II Pro da Xilinx) e os experimentos de avaliação realizados demonstraram que a solução proposta não introduz um sobrecusto considerável.

Sepúlveda, Strum e Chau [28] propuseram o uso de uma técnica de chaveamento híbrida em uma NoC combinando chaveamento por circuito com chaveamento por pacotes para evitar ataques de DoS e garantir disponibilidade para tráfego de informações críticas sem reserva de recursos. No mecanismo proposto, as mensagens são classificadas por políticas de segurança como críticas ou não críticas. As primeiras são transferidas utilizando-se chaveamento por circuito, enquanto as últimas são transferidas com o uso de chaveamento por pacotes. As políticas de segurança foram implementadas adicionando funcionalidades ao roteador e alterando alguns parâmetros nas configurações locais da NoC. Os experimentos de avaliação foram realizados com o uso de um ambiente de simulação SystemC TLM. Para efeito de comparação, foram implementadas quatro NoCs com topologia em malha 2D 4×4: (*i*) com chaveamento por pacote e reserva de canais virtuais; (*ii*) com chaveamento por circuito; (*iii*) com multiplexação por divisão de tempo; e (*iv*) com chaveamento híbrido (circuito e pacotes). Foram avaliados o impacto na eficiência e o desempenho das quatro implementações. Nos experimentos, a taxa de dados críticos foi definida segundo três configurações diferentes: 30%, 50% e 70% do total do tráfego. Os resultados demonstraram que o mecanismo proposto evita contenção na rede, reservando recursos para mensagens críticas sem negar serviço de comunicação às mensagens não críticas. A avaliação mostrou que a solução proposta apresenta eficiência de 98% de garantia contra ataques de DoS. O consumo de energia e a latência na rede melhoraram até 62% e 55%, respectivamente, quando comparados com os da NoC de melhor esforço.

Porquet, Schwarzt e Greiner [29] propuseram uma solução para a implementação de uma arquitetura segura para o compartilhamento de memória em NoC. Um componente na interface de rede atua como um *firewall*, filtrando o acesso com base em um identificador

denominado CID (*Compartment IDentifier*) e na transação requisitada no endereço de memória usando uma tabela de permissão. O trabalho não reportou resultados de avaliação.

Kapoor *et al*. [30] apresentaram um *framework* de autenticação criptografada para sistemas baseados em NoC implementado na interface de rede de forma a prover segurança na comunicação entre núcleos. Os núcleos seguros podem se comunicar utilizando chaves permanentes, enquanto a comunicação entre núcleos seguros e núcleos não seguros é feita utilizando chaves de sessão temporárias. Um contador limitador de tráfego é usado para evitar ataques de DoS que visam consumir largura de banda. Além disso, tabelas de direito de acesso evitam acessos não autorizados à memória. O *framework* proposto foi modelado usando Verilog/VHDL, sintetizado em FPGA e simulado usando o emulador NoCem [31]. Segundo os autores, os resultados mostram que o *framework* proposto adiciona um sobrecusto de silício tolerável e não impactou no desempenho da rede, exceto por alguma latência inicial.

Baron, Wangham e Zeferino [32] propuseram uma solução para proteger uma NoC de ataques de DoS, a qual foi implementada como um *wrapper* de *hardware* posicionado entre a interface de rede e o roteador. Esse *wrapper*, denominado SEW (*SEcurity Wrapper*), realiza a filtragem dos fluxos de comunicação injetados na rede, descartando pacotes que comprometam a sua disponibilidade ou regulando a taxa de injeção de fluxos que pretendam consumir uma largura de banda maior do que a que foi prevista pelo projetista do SoC. O *wrapper* foi descrito em VHDL e sintetizado na tecnologia SAED 90nm CMOS da Synopsys. Para avaliação do impacto no desempenho e da efetividade da solução implementada, foram realizados experimentos no ambiente de simulação BrownPepper [33], no qual a rede SoCIN é modelada em SystemC RTL (*Register-transfer Level*). Os resultados obtidos demonstraram que a solução proposta é efetiva em proteger a rede dos ataques considerados, apresenta baixo impacto no desempenho da NoC e um pequeno sobrecusto de área de silício (4,1%) e de potência dissipada (2,5%) em relação aos custos do roteador da rede.

5 Análise Comparativa

Esta seção apresenta uma comparação dos trabalhos apresentados na seção anterior. Conforme sumarizado no Quadro 2, para cada trabalho analisado, identifica-se o ano de publicação, o tipo de ataque abordado, a propriedade de segurança tratada, o mecanismo de segurança adotado e o componente do SoC ou da NoC no qual foi implementada a solução proposta. O quadro ainda contabiliza o número de ocorrências de cada alternativa para esses itens no universo de trabalhos analisados. Alguns aspectos devem ser destacados em relação a esse quadro comparativo. O ataque ativo de repetição e a propriedade de não repúdio não foram incluídos porque não foram considerados em nenhum dos trabalhos analisados. A categoria de ataque de liberação de conteúdo da mensagem incluiu os ataques que fazem acesso à memória para obter informações sem a devida autorização. A componente Interface

de Rede incluiu também implementações feitas na forma de *wrappers* colocados entre o núcleo e o roteador da rede.

Quadro 2. Quadro comparativo: ataques, propriedades, mecanismos e componentes

Trabalho	Ano	Tipo de Ataque					Propriedade de Segurança				Mecanismo de Segurança					Componente		
		Liberação de conteúdo	Análise do tráfego	Disfarce	Modificação de mensagem	Negação de serviço	Autenticação	Disponibilidade	Confidencialidade	Integridade	Controle de acesso	Cifragem	Integridade dos dados	Troca de informações de autenticação	Controle de roteamento	Núcleo	Interface de Rede	Roteador
[18]	2003	X		X	X		X		X	X	X	X	X	X		X	X	
[19]	2007	X			X	X	X	X		X	X		X				X	X
[21]	2007	X			X	X		X	X	X	X		X				X	X
[22]	2008	X			X		X		X	X	X		X				X	
[10]	2008	X			X	X	X	X	X		X		X				X	
[9]	2009	X			X	X	X	X	X	X	X		X				X	X
[23]	2009	X	X						X						X		X	
[25]	2010	X			X				X	X	X		X				X	
[26]	2010				X	X		X		X	X		X				X	
[28]	2010					X		X							X			X
[29]	2011	X			X				X	X	X						X	
[30]	2013	X				X	X	X	X		X	X		X			X	
[32]	2013					X		X			X						X	
Ocorrências		10	1	1	9	8	6	8	9	8	11	2	8	2	2	1	12	4

Pela análise do quadro comparativo, pode-se identificar que: (*i*) os principais ataques tratados pelas pesquisas sobre segurança em NoCs são os de modificação de mensagem, de liberação do conteúdo da mensagem e de negação de serviço; (*ii*) as propriedades de segurança mais abordadas são as de integridade, de confidencialidade e de disponibilidade;

(*iii*) os mecanismos mais utilizados são os de controle de acesso e de integridade dos dados; e (*iv*) a interface de rede é o componente mais utilizado na implementação das soluções para provimento de segurança.

No Quadro 2, não foram incluídas informações sobre impacto das soluções implementadas no custo e no desempenho do sistema ou da NoC. Isso porque nem todos os trabalhos apresentam tais informações e, além disso, os que apresentam baseiam-se em diferentes técnicas, tecnologias e métricas, o que dificulta a organização desses dados em um único quadro e a comparação dos seus resultados. Porém, é possível sumarizar que as implementações reportadas utilizaram linguagens de descrição de *hardware* (VHDL ou Verilog) ou de sistema (SystemC) para modelar as soluções propostas, sendo que algumas dessas soluções também foram sintetizadas para tecnologias de circuito integrado do tipo FPGA ou ASIC (*Application-specific Integrated Circuit*). Com relação às métricas de avaliação, foram considerados o subrecusto de área, o sobrecusto de energia e o impacto da solução proposta no desempenho da NoC. Ressalta-se que, na maioria dos trabalhos analisados, as soluções propostas apresentaram baixo impacto no desempenho do sistema e um sobrecusto de área e de energia aceitável, especialmente se for considerado o benefício obtido com a melhoria da segurança do sistema.

6 Conclusões

Este artigo apresentou uma visão sobre o cenário da pesquisa a respeito de segurança em NoCs, cenário este que se torna cada vez mais relevante com o aumento da quantidade de núcleos em sistemas integrados e o uso desses sistemas em um número cada vez maior de aplicações. Como a NoC é a melhor solução de interconexão para tais sistemas e gerencia as comunicações entre os seus núcleos, investir em soluções de segurança no nível da rede significa proteger não apenas a NoC, mas o sistema, a aplicação e o seu usuário.

A manipulação de dados críticos dentro da NoC faz com que seja de grande importância o atendimento dos requisitos de segurança. Esses requisitos podem ser caracterizados pela arquitetura de segurança OSI, a qual engloba ataques, mecanismos e propriedades de segurança. O emprego de mecanismos de segurança garante que as propriedades de segurança não sejam afetadas pelos ataques.

Conforme visto, a pesquisa sobre segurança em NoCs tem focado seus esforços nos ataques de liberação de conteúdo da mensagem, de modificação de mensagem e de negação de serviço. Porém, além desses, outros tipos de ataque passam a se tornar relevantes e demandam por soluções para proteger a rede e o sistema. Por exemplo, os ataques de análise de tráfego podem ser utilizados em processos de engenharia reversa. A implementação de contramedidas para esse tipo de ataque pode ajudar a proteger a propriedade intelectual do sistema e/ou dos seus núcleos. Esses e outros problemas representam, por consequência, novas oportunidades de investigação para pesquisadores da área.

7 Agradecimentos

Este estudo foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq e pelo Instituto Nacional de Ciência e Tecnologia de Sistemas Micro e Nanoeletrônicos – INCT NAMITEC.

Referências

[1] G. E. Martin and H. Chang, *Winning the SoC Revolution: Experiences in Real Design*, New York: Springer, 2003.

[2] N. E. Jerger and L. Peh, *On-Chip Networks*, Synthesis Lectures on Computer Architecture, Morgan & Claypool, 2009.

[3] S. K. Tewksbury, M. Uppuluri and L. A. Hornak, "Interconnections/micro-networks for integrated microelectronics," *in Global Telecommunications Conf. (GLOBECOM 1992)*, Orlando, USA, Dec. 6-9, 1992, pp.180-186.

[4] P. Guerrier and A. Greiner, "A Scalable Architecture for System-on-Chip Interconnections," *in: 2nd Sophia Antipolis Micro-Electronics Conf. (SAME 1999)*, Sophia Antipolis, France, 1999, pp. 90-93.

[5] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Öberg, M. Millberg, D. Lindqvist, "Network on Chip: an Architecture for Billion Transistor Era," *in 18th IEEE NorChip Conf. (NORCHIP 2000)*, Turku, Finland, Nov. 2000.

[6] L. Benini and G. De Micheli, "Networks on Chips: a new SoC Paradigm," *Computer* , vol. 35, no. 1, pp. 70,78, Jan. 2002.

[7] C. A. Zeferino, A. A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip," *in 16th Symp. on Integrated Circuits and Systems Design (SBCCI 2003)*, São Paulo, Brazil, Sep. 8-11, 2003, pp. 169-174.

[8] M. Texier, Network-*on-Chip Security: Overview of Existing Solutions*, University of South Brittany. [Online]. Disponível em: http://www.mtexier.com

[9] M. J. Sepúlveda, M. Strum and W. J. Chau, "Performance Impact of QoSS (Quality-of-Security-Service) Inclusion for NoC-based Systems," *in 17th Int. Conf. on Very Large Scale Integration (VLSI-SoC 2009)*, Florianópolis, Brazil, Oct. 12-14, 2009.

[10] L. Fiorin, G. Palermo and C. Silvano, "A Security Monitoring Service for NoCs," *in 6th IEEE/ACM/IFIP Int. Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS 2008)*, Grenoble, France, Oct. 11-16, 2008, pp. 197-202.

[11] G. De Micheli and L. Benini. *Networks on chip: Technology and Tools*. New York: Elsevier North-Holland, 2006.

[12] C. A. Zeferino, *Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho*, Programa de Pós-Graduação em Computação, UFRGS, Porto Alegre, Brazil, 2003.

[13] C. E. Landwehr, "Computer Security," *Int. J. of Information Security (IJIS)*, New York, vol. 1, n. 1, pp. 3-13, Aug. 2001.

[14] Associação Brasileira de Normas Técnicas, NBR ISO/IEC 27001:2006: *Sistema de Gestão de Segurança da Informação*, Rio de Janeiro, 2006.

[15] J. E. M. S. Brandão and J. S. Fraga, "Gestão de Riscos de Segurança," *in* C. A. Maziero, *Livro Texto dos Minicursos do SBSeg 2008*, Porto Alegre: SBC, 2008

[16] National Information Assurance Glossary, *CNSSI-4009: Committee on National Security Systems*, 2006.

[17] W. Stallings, *Criptografia e Segurança de Redes: Princípios e Práticas*, 4.ed., Prentice Hall, 2008.

[18] C. Gebotys and R. J. Gebotys, "A Framework for Security on NoC Technologies," *in IEEE Computer Society Annual Symp. on VLSI (ISVLSI 2003)*, Tampa, USA, Feb. 20-21, 2003, pp. 113-117.

[19] J. Diguet, S. Evain, R. Vaslin, G. Gogniat and E. Juin. "NoC-centric Security of Reconfigurable SoC," in *1st Int. Symp. on Networks-on-Chip (NOCS 2007)*, Princeton, USA, May 7-9, 2007, pp. 223-232.

[20] S. Evain, J. Diguet and D. Houzet, "μSpider: a CAD Tool for Efficient NoC Design," *in 22nd IEEE NorChip Conf. (NORCHIP 2004),* Oslo, Norway, Nov. 2004, pp. 218-221.

[21] L. Fiorin, C. Silvano and M. Sami, "Security aspects in Networks-on-Chips: Overview and Proposals for Secure Implementations", *in 10th Euromicro Conf. on Digital System Design Architectures, Methods and Tools (DSD 2007)*, Lübeck, Germany, Aug. 29-31, 2007, pp. 539-542.

[22] L. Fiorin, G. Palermo, S. Lukovic, V. Catalano and C. Silvano, "Secure Memory Accesses on Networks-on-Chip", *IEEE Trans. on Computers*, Washington, vol. 57, n. 9, pp. 1216-1229, Sep. 2008.

[23] R. Stefan and K. Goossens, "NoC Security using Multipath Routing," *in 20th Workshop on Circuits, Systems and Signal Processing (ProRISC 2009)*, Velhoven, The Netherlands, Nov. 2009, pp. 522-525.

[24] K. Goossens, J. Dielissen and A. Radulescu, "A Æthereal Network on Chip: Concepts, Architectures, and Implementations," *IEEE Design & Test of Computers*, pp. 414-421, 2005.

[25] T. Huffmire, T. Levin, T. Nguyen, C. Irvine, B. Brotherton, G. Wang, T. Sherwood and R. Kastner, "Security Primitives for Reconfigurable Hardware-based Systems," *ACM*

Trans. on Reconfigurable Technology and Systems, vol. 3, n. 2, pp. 10:1-10:35, May. 2010.

[26] S. Lukovic and N. Christianos, "Hierarchical Multi-agent Protection System for NoC based MPSoCs," *in Int. Wksp. on Security and Dependability for Resource Constrained Embedded Systems (S&D4RCES 2010)*, Vienna, Austria, Sep. 14, 2010. 7p.

[27] S. Lukovic, P. Pezzino and L. Fiorin, "Stack Protection Unit as a Step Towards Securing MPSoC," *in IEEE Int. Symp. on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW 2010)*, Atlanta, USA, Apr. 19-23, 2010. pp. 1-4.

[28] M. J. Sepúlveda, M. Strum and W. J. Chau, "An Hybrid Switching Approach for NoC-based Systems to avoid Denial-of-Service SoC Attacks," *in 16th Iberchip Wksp (IWS 2010)*, Iguazu Falls, Brazil, Feb. 23-25, 2010.

[29] J. Porquet, C. Schwarzt and A. Greiner, "NoC-MPU: a Secure Architecture for Flexible co-hosting on Shared Memory MPSoCs", *in Design, Automation & Test in Europe Conf. & Exhib. (DATE 2011)*, Grenoble, France, Mar. 14-18, 2011, pp. 1-4.

[30] H. K. Kapoor, G. B. Rao, S. Arshi and G. Trivedi, "A Security Framework for NoC Using Authenticated Encryption and Session Keys," *Circuits, Systems, and Signal Processing*, vol. 32, n. 6, pp. 2605-2622, Dec. 2013.

[31] G. Schelle and D. Grunwald, *NoCem User Guide and Release Documentation*, April 2007.

[32] S. Baron, M. S. Wangham and C. A. Zeferino, "Security Mechanisms to Improve the Availability of a Network-on-Chip," *in IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS 2013)*, Abu Dhabi, UAE, Dec. 8-13, 2013, pp. 609-612.

[33] J. V. Bruch, M. R. Pizzoni and C. A. Zeferino, "BrownPepper: A SystemC-based simulator for performance evaluation of Networks-on-Chip," *in 17th Int. Conf. on Very Large Scale Integration (VLSI-SoC 2009)*, Florianópolis, Brazil, Oct. 12-14, 2009, pp. 223-226.