UM ESTUDO DOS ALGORITMOS DE CRIPTOGRAFIA LEVE PARA DISPOSITIVOS IOT

A STUDY OF LIGHTWEIGHT ENCRYPTION ALGORITHMS FOR IOT DEVICES

José dos Santos Machado
Mestre em Ciência da Computação e Técnico de Tecnologia da Informação do Instituto Federal de Sergipe (IFS). E-mail: jsmac18@hotmail.com

Danilo Souza Silva
Mestre em Ciência da Computação pela Universidade Federal de Sergipe (UFS). E-mail: danilo.silva@dcomp.ufs.br

Adauto Cavalcante Menezes
Mestre em Ciência da Computação e Técnico de Tecnologia da Informação do Instituto Federal de Sergipe (IFS). E-mail: adauto.cavalcant@gmail.com

Edward David Moreno Ordonez
Doutor em Engenharia Elétrica e Professor da Universidade Federal de Sergipe (UFS). E-mail: edwdavid@gmail.com

Admilson de Ribamar Lima Ribeiro
Doutor em Engenharia Elétrica e Professor da Universidade Federal de Sergipe. E-mail: admilson@ufs.br

Resumo: Dispositivos IoT estão cada vez mais presentes em diversas áreas das atividades humanas, coletando, processando, armazenando e compartilhando informações sensíveis sobre seus usuários. Contudo, devido às dimensões físicas reduzidas e limitação de recursos computacionais desses dispositivos, implementar algoritmos tradicionais para prover segurança torna-se uma tarefa desafiadora. Para superar essa limitação, algoritmos de criptografia leve foram propostos. Esses tipos de algoritmo são adaptados para implementação em ambientes restritos, incluindo etiquetas RFID, sensores, cartões inteligentes, dispositivos de cuidados de saúde, etc. Estudos de implementações de soluções de segurança em ambiente de *hardware* limitado foram realizados com algoritmos criptográficos conhecidos. Porém, existem na literatura inúmeras cifras de criptografia com especificações variadas e, se a escolha de um algoritmo de criptografia não for adequada, pode afetar diretamente fatores determinantes para o funcionamento do dispositivo, como o tempo de vida da bateria, memória do *hardware*, latência computacional e largura de banda na comunicação dos dados. Nesse contexto, este trabalho tem o objetivo de apresentar algoritmos de criptografia leve desenvolvidos para dispositivos de recursos limitados citados na literatura recente. Por fim, são apresentadas direções futuras para trabalhos de pesquisas.
Palavras-chave: Criptografia. Cifras Leve. Dispositivos de Recursos Limitados. Internet das Coisas.

Abstract: IoT devices are increasingly present in several areas of human activities, collecting, processing, storing, and sharing information about their users. However, due to the reduced physical dimensions and limited computational resources of these devices, implementing traditional algorithms to provide security becomes a challenging task. To overcome this limitation, lightweight encryption algorithms have been proposed. These types of algorithms are adapted for implementation in restricted environments, including RFID tags, sensors, smart cards, healthcare devices, etc. Studies of implementations of security solutions in a limited hardware environment were carried out with known cryptographic algorithms. However, there are lots of cryptographic figures in the literature with varied specifications, and if the choice of an encryption algorithm is not suitable, this can directly affect the determining factors for the functioning of the device, such as battery life, memory of the hardware, computational latency, and bandwidth in

data communication. In this context, this work has the objective to present lightweight cryptography algorithms developed for limited resource devices mentioned in the recent literature. Finally, they are recruited for future research work.
Keywords: Cryptography. Lightweight Ciphers. Limited Resources Devices. Internet of Things.

INTRODUÇÃO

A Internet das Coisas (IoT) promete ser a próxima grande revolução da *World Wide Web*. Atualmente, aplicações como o monitoramento ambiental, transporte inteligente, bem como sistema de monitoramento de saúde e casas inteligentes agregam diversos benefícios na vida das pessoas (LEE; LEE, 2015; CHEN et al., 2014).

A IoT representa uma interligação de um grande número de dispositivos inteligentes com baixo recursos por WSN (*Wireless Sensor Network*), que dispõe de uma grande rede de sensores e atuadores com restrita capacidade de processamento e memória. Prover recursos de segurança eficientes para garantir confiabilidade e privacidade dos usuários são importantes desafios para a popularização da IoT (PAWAR; AGARWAL, 2017; BAE; SHIN, 2016).

No entanto, apesar da IoT estar revolucionando o mundo com os mais diversos projetos, também está causando perigo aos dados e às pessoas, já que a segurança não está andando alinhada ao desenvolvimento de novos produtos (BAE; SHIN, 2016; BHARDWAJ et al., 2017). A integração com a internet implica que os dispositivos terão um identificador exclusivo. Esses dispositivos herdaram ameaças de segurança de um computador interligado à internet, pois possuem capacidade de processamento e comunicação na rede (KONG et al., 2015).

A segurança é um grande desafio e garantir um nível adequado de proteção aos dados é um problema crítico para dispositivos com baixos recursos computacionais (BUCHMANN et al., 2016; MOHD et al. 2015). Não obstante, mecanismos de segurança precisam ser fornecidos para que essas informações não sejam acessadas por pessoas não autorizadas e algoritmos de criptografia são usados para garantir a confidencialidade, já que os atacantes não podem interpretar o texto cifrado que é enviado (JING et al., 2014; MCCANN et al., 2015; MOHD et al., 2015; SUNDARAM et al., 2015).

O problema reside na busca de um algoritmo de cifra adequado que funcione e se encaixe confortavelmente dentro do ambiente de *hardware* limitado, observando consumo de energia que é uma forte restrição que afeta o tempo de vida dos dispositivos na rede. Geralmente, algoritmos de criptografias direcionados para dispositivos com poucos recursos são referidos como algoritmos de cifras leves (MCCANN et al., 2015; MOHD et al., 2015; KUSHWAHA et al., 2014).

Os algoritmos de criptografias de dados são divididos em duas categorias: algoritmo de criptografia simétrica, que contém uma chave privada, e algoritmo de criptografia assimétrica, que contém uma chave pública e outra privada. Devido à complexidade computacional, a elevada utilização de memória e o alto consumo de energia, geralmente os algoritmos de criptografias assimétricos não são utilizados para implementações em *hardwares* limitados (BUCHMANN et al., 2016; JING et al., 2014). Além disso, o objetivo do algoritmo deve ser o de assegurar a criptografia e integridade dos dados nos sensores que têm memória limitada e poder de processamento restrito (SUNDARAM et al., 2015).

Nesta direção, este artigo contribui com uma pesquisa bibliográfica sobre soluções de algoritmos de criptografias leves simétricos usados em IoT para dispositivos de recursos limitados que fazem uso

da área de rede de sensores sem fio (WSN). Esta pesquisa coleta informações da literatura recente e vai auxiliar pesquisadores na área da implementação de criptografia. Além disso, visa, também, contribuir para uma melhor compreensão dos requisitos e tendências da atual abordagem sobre criptografia na área das cifras leves de blocos simétricos.

O artigo está estruturado da seguinte forma: a seção 2 apresenta a taxonomia em relação ao tema proposto, seguido da seção 3, que explana a metodologia utilizada na revisão sistemática. A seção 4 analisa os trabalhos relacionados que foram tomados como referências para este artigo e na seção 5 são apresentados alguns algoritmos de cifras leves. A seção 6 apresenta as conclusões e considerações finais do presente artigo e finalizamos apresentando futuras direções de trabalhos acadêmicos com relevância no meio científico.

TAXONOMIA

Novos dispositivos de rede que constituem a IoT são de baixo consumo energético e possuem capacidades limitadas de recursos computacionais (DORRI et al., 2017). Esses dispositivos devem dedicar a maior parte da energia disponível e recursos computacionais para executar as funcionalidades principais da aplicação. Contudo, é inegável que a seleção de um algoritmo de criptografia adequado pode afetar dinamicamente o tempo de vida e desempenho de um dispositivo em termos de consumo da bateria, memória do *hardware*, latência computacional e largura de banda de comunicação. Quando os recursos de *hardware* são escassos, a utilização cuidadosa desses recursos é essencial.

Para selecionar um algoritmo criptográfico apropriado, otimizado para uma aplicação ou um ambiente IoT, o entendimento de requisitos em termos de *hardware* e as especificações da plataforma destinada ao desenvolvimento são obrigatórias. Com a compreensão dos diferentes lados, desenvolvedores e pesquisadores são capazes de escolher uma solução adequada para todas as classes de *hardware*, tendo em consideração as exigências de segurança (KONG et al., 2015).

A classificação da implementação de cifra é uma tarefa árdua e alguns dos pesquisadores a descrevem como encontrar uma agulha no palheiro (MOHD et al., 2015). A razão principal é que existe um considerável número de artigos de pesquisas publicados na implementação de cifra, abordando diversos problemas como desempenho, taxa de transferência e consumo de energia. Enquanto o espaço de implementação é restrito, com o avanço na tecnologia, essa classificação torna-se dinâmica e pode ser modificada constantemente ao longo do tempo. Além disso, diferentes sistemas de classificação, a exemplo do desempenho ou plataformas, produzem diferentes taxonomias (MOHD et al., 2015).

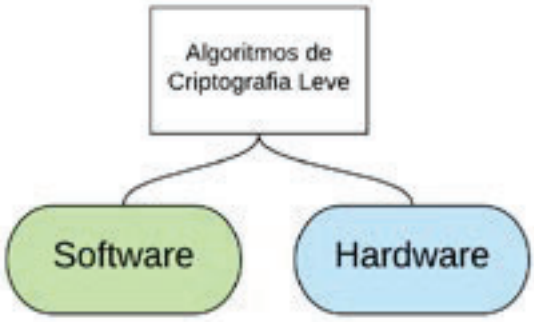
Em 2017, o Instituo Nacional de Padrões e Tecnologia ((National Institute of Standards and Technology – NIST) publicou o "*Report on Lightweight Cryptography*", abordando métricas de avaliação para o grupo de cifra que inclui os requisitos para definir que um algoritmo pode ser considerado um algoritmo de criptografia de classe "leve". Contudo, muitas das propostas de cifra afirmam que o algoritmo proposto é leve, mas os fundamentos de tais alegações não são claros. A motivação deve ser específica para a implementação de dispositivos quando os ambientes de recursos limitados (WSN em particular), estiverem no centro do foco da pesquisa, uma vez que o objetivo é encontrar uma solução criptográfica que utilize o mínimo de recurso possível.

Investigar os dados reportados de estudos

existentes e suas conclusões pode levar a erros de interpretação, como observado por pesquisadores (MOHD et al., 2015). Os relatórios são usados para extrair observações intuitivas e desenhar um mapa das melhores cifras realizadas em várias categorias de métricas.

Os artigos de pesquisas para implementações de cifras leves são geralmente classificados para a implementação de *software* e *hardware*. Uma vez que o foco da pesquisa é sobre a implementação de dispositivo de baixo recurso, a classificação das cifras foi analisada no contexto de seu desempenho de sistema e plataforma.

Figura 1 - Taxonomia



Fonte: Os autores.

A taxonomia ilustrada na Figura 1 representa a classe de cifras denominada de "bloco leve". Algumas implementações abrangem várias classes e, em alguns casos, não é fácil para classificar uma implementação em particular. Por exemplo, um código de *software* de cifra escrita em C pode não ser otimizado pelo compilador, portanto, a sua dependência da máquina é óbvia. No entanto, uma estrutura de classificação deve ajudar na compreensão do espaço de sua implementação.

METODOLOGIA

Para alcançar o grau de rigor científico, procurou-se assegurar o processo de investigação a partir das concepções de uma Revisão Sistemática (RS). A revisão sistemática de literatura identifica, avalia e interpreta todas as pesquisas disponíveis relevantes para responder uma questão específica, área temática ou fenômeno de interesse (KITCHENHAM, 2015).

A importância do estabelecimento de um processo de RS se concretiza ao observar a definição das fases: a) Planejamento; b) Procedimentos de condução extração da RS e, ao final, c) Procedimentos de elaboração de relatórios da RS.

Dessa forma, os resultados são mais confiáveis em relação à revisão de literatura primária em virtude de sua forma rigorosa, que dá possibilidades de repetição e auditagem. Nessa direção, a RS teve o seu foco na classificação de estudos de algoritmos de criptografias leves, baseado em dispositivos de recursos limitados para IoT.

A fim de estabelecer os interesses da pesquisa, o protocolo construído definiu os seguintes critérios de seleção de trabalhos: artigos publicados no período de 2013 a 2018; artigos completos disponíveis online; idioma inglês; estudos que abordem algoritmos de criptografia leve para dispositivos IoT.

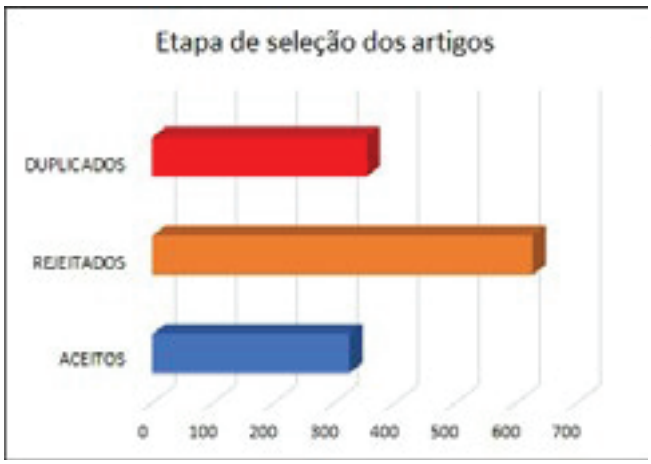
Na primeira etapa da revisão foram identificados 1.305 artigos nas bases de dados consultadas (ACM (237 artigos), IEEE Xplorer (311 artigos), *Science Direct* (280 artigos), *Scopus* (282 artigos) e *Springer* (195 artigos)). A Figura 2 apresenta melhor visualmente os resultados das pesquisas nas bases de consultas.

Figura 2 - Resultado das bases

Fonte: Os autores.

Na segunda etapa, houve o julgamento através dos critérios de exclusão dos 1.305 trabalhos relacionados ao processo de identificação. 325 foram aceitos para serem avaliados na etapa de extração da RS, 980 trabalhos foram excluídos, sendo 627 pelos critérios definidos no protocolo de exclusão (trabalhos que não apresentaram texto completo; trabalhos que os termos de busca não se apresentam nos campos: título, resumo e palavras-chave, e trabalhos de conclusão de curso). Além disso, 353 trabalhos foram excluídos por estar em duplicidade com artigos das outras bases. A Figura 3 apresenta o resultado da etapa de seleção dos artigos.

Figura 3 - Etapa de seleção



Fonte: Os autores.

Dos 325 trabalhos aceitos por pelo menos um critério de seleção, esses foram conduzidos à etapa de extração. A etapa possibilitou a seleção e condução de 37 trabalhos, os quais atenderam os critérios formalizados no protocolo da revisão sistemática da literatura.

TRABALHOS ANALISADOS

Para a realização deste artigo, foram analisados trabalhos publicados nas plataformas selecionadas e que possuíam referências com o tema tratado. Foi identificado que alguns artigos não tinham uma relação direta com o tema em pesquisa, os blocos de cifras leves, mas que citavam de segurança de forma mais genérica, voltada para IoT.

Buchmann et al. (2016) propuseram uma variante da estrutura de esquema de criptografia R-LWEEnc em que se substituiu a distribuição de ruído Gaussiana com uma distribuição binária uniforme, mostrando como lidar com o erro não-simétrico durante a descriptografia. Percebeu-se que os resultados demonstram ser bastante satisfatórios para pequenos dispositivos. O trabalho buscou comparar uma implementação para um ARM e um microcontrolador AVR, com implementações de microcontroladores existentes de criptografia de chave pública. Devido ao maior nível de otimização, a implementação ARM se mostrou mais rápida do que a implementação AVR. Além disso, armazenando dois coeficientes chave em uma palavra de dados de 32-bits, em vez de um coeficiente de uma palavra de dados de 8-bits, o requisito de memória para o armazenamento de chave foi dobrado.

Já Jing et al. (2014) buscaram fazer uma comparação entre segurança em redes tradicionais e em IoT. O trabalho focou na arquitetura de segurança e questões de segurança relativas à

IoT, que foi dividida em três camadas: camada de recepção, camada de transporte e camada de aplicação. Baseada nessa divisão, o trabalho analisa as características e questões de segurança de cada camada. Percebeu-se que a segurança relativa à IoT é bem mais crítica, principalmente quando se considera a comunicação dos sensores sem fio direto para a internet, por conta da limitação de recursos como processamento, memória e autonomia de carga. Ao comparar as questões de segurança entre a Internet das Coisas e a rede tradicional, concluiu-se que o sistema de IoT é implementado em um ambiente mais perigoso, por conta das limitações e recursos reduzidos de rede. A busca por soluções leves seria sempre a primeira opção para a segurança da Internet das Coisas.

Kneževic et al. (2012) fizeram um estudo no qual realizaram uma comparação entre o desempenho de cifras de blocos leves e algoritmos de baixas latências, tentando identificar quais os algoritmos leves que poderiam ser convertidos para baixa latência. No artigo é citado que o advento da RFID e outras tecnologias sem fio provocaram um aumento no interesse por estruturas de baixo consumo de energia e criptografia de baixo custo. As primeiras a serem exploradas foram as cifras de fluxos como, por exemplo, no projeto *eSTREAM*, seguido de todo um conjunto de cifras de blocos leves, como o *tea, noekeon, mini-aes, mcrypton, sea, hight, desxl, clefia, present, mibs, katan/ktantan, printcipher, klein, led, piccolo* e outros. O campo foi recentemente ampliado pela introdução de várias novas funções *hash* de baixo custo, como *dmpresent, keccak-f[400]/-f[200], quark, photon e spongent*. Percebe-se, durante o transcorrer do artigo, que os resultados são fortemente influenciados pelas propriedades das cifras, mais especificamente, o número e a complexidade dos *loops*, bem como a similaridade dos processos de criptografia e descriptografia, os quais têm uma influência significativa sobre o desempenho do algoritmo.

No artigo de McCann et al. (2015) foram feitas experimentações com um microcontrolador ARM Cortex-M4, visando avaliar o consumo de energia do uso de diferentes contramedidas SCA em dispositivos embarcados 32 bits. No estudo, foram usadas as cifras de blocos leves *PRESENT, KLEIN e ZORRO*. O trabalho demonstrou que a otimização para a velocidade levará as cifras a serem mais eficientes no que se refere ao consumo de energia. Já as cifras otimizadas vão exigir um consumo adicional de energia para criptografar e descriptografar.

O artigo de Mishra (2015) fez um estudo conceitual com relação aos desafios gerais enfrentados pela IoT, dando destaque à segurança através dos algoritmos criptográficos leves. O artigo tabula deficiências relativas ao presente esquema de criptografia e como elas dificultam o trabalho nos dispositivos com recursos limitados. Segundo Mishra, a solução é a criação de um protocolo de segurança de cifra leve, visando garantir a confidencialidade e privacidade para o usuário.

Wu e Han (2013) fizeram uma análise que tinha como base a IoT, descrevendo os conceitos das tecnologias de computação confiáveis sobre módulos de criptografia de confiança (MCC), mostrando a hierarquia básica e um modelo de segurança baseado em um sistema de IoT simples. Com base nesse modelo de segurança, foi criada uma solução para IoT usando o tecnologia TCM-base que abrange a inicialização segura, o armazenamento seguro, as métricas de plataforma e relatórios confiáveis. Após os resultados da análise a solução pode fornecer salvaguarda de segurança confiável para o seu desenvolvimento e na sua aplicação.

Já Sundaram et al. (2015) propuseram a utilização de criptografia com algoritmos *hash* para garantir uma forma segura de enviar mensagens entre os dispositivos embarcados, visando a confidencialidades das informações trafegadas. O trabalho discute os algoritmos 5-RC, Skipjack e AES. Segundo os autores, a utilização do algoritmo *hash* demonstrou que pode ser aplicada para redes de sensores sem fio, pois faz uso de operações de baixa complexidade, não estando sujeito a ataques de força bruta, devido a seu comprimento da chave de 128 bits. Segundo eles, o algoritmo de integridade pode ser usado para dispositivos de baixa potência, demonstrando-se altamente seguro, eficiente e rápido quando comparado com os algoritmos existentes.

Os outros dois artigos analisados correspondem a *surveys* que tratam do estudo de algoritmos para dispositivos de baixo recurso. O estudo de Mohd et al. (2015) fez um levantamento detalhado com relação aos tipos de métricas de performance de *software* e *hardware*, discutindo seus resultados. Além disso, estuda as plataformas de implementação de *software* e as plataformas de implementação de *hardware* de forma detalhada, tabulando os algoritmos utilizados nessas implementações, com suas respectivas características. No final, o artigo faz uma breve discussão com relação às questões de pesquisas em aberto, mais especificamente com relação a modelos de performance, *hardware* trojans, métricas de segurança e estilo de códigos de programação. Por outro lado, Sehrawa e Gill (2018) apresentaram um estudo comparativo entre várias cifras de bloco leves adequadas para aplicações de IoT, juntamente com seus benefícios e limitações. Os autores também apresentam uma proposta de trabalhos futuros para o desenvolvimento de novas cifras leves.

Por fim, Kong et al. (2015) fizeram uma pesquisa abrangente de soluções de criptografia simétrica modernas para ambientes com recursos limitados. Inicialmente, o artigo descreve os ambientes com restrições de recursos envolvendo (RRE) e que tem impacto pela utilização dessas soluções de criptografia, como: *Wireless Sensor Network* (WSA), *Radio Frequency Identification* (RFID), *Wireless Identification and Sensing Platform* (WISP) e *Internet of Things* (IoT). Após essa identificação, os algoritmos são divididos em quatro grupos: algoritmos de cifras de blocos de chaves simétricas modernas, *Involution cipher*, algoritmos de cifras leves e algoritmos de cifras de fluxos. Para cada grupo são descritos e analisados, de forma minuciosa, vários algoritmos, com suas características e tabelas comparativas, dando uma visão ampla com relação ao que existe no mercado.

ALGORITMOS DE CRIPTOGRAFIA LEVE

Recentemente, várias implementações de *software* e *hardware* de cifras leves foram projetadas para aplicações IoT. A seguir, é apresentado uma breve descrição de alguns dos principais algoritmos de cifras leves encontrados no mercado de sistemas embarcados. A Tabela 1 apresenta essas cifras de bloco leve otimizadas para implementações de *hardware* e *software*, ordenadas pelo ano de criação e apresentado o tamanho da chave e tamanho do bloco em bits.

Tabela 1: Algoritmos de cifra leve

Cifra	Ano	Tamanho da Chave (bits)	Tamanho do Bloco (bits)
NOKEON	2000	128	128
CLEFIA	2007	128/192/256	128
PRESENT	2007	80/128	64
KLEIN	2010	64/80/96	64
LED	2011	64/128	64
SIMON	2012	64/ 72/ 96/ 128/ 144/ 192/ 256	32/48/64/96/128
SPECK	2012	32/ 64/ 72/ 96/ 128	64/ 72/ 96/ 128/ 144/ 192/ 256
Edib	2013	80/128	64
LEA	2013	128/192/256	128
Chaskey	2014	128	128
HISEC	2014	80	64
ITUBee	2014	80	80
LAC	2014	80	64
OLBCA	2014	80	64
PICO	2015	128	64
RECTANGLE	2015	80/128	64
LAX	2016	128/256	64/128
SPARX	2016	128/256	64/128
Lilliput com EGFN	2016	80	64
MANTIS	2016	128	64
SKINNY	2016	64-384	64/128
RoadRunneR	2016	80/128	64
DLBCA	2017	80	32
GIFT	2017	128	64/128
LiCi	2017	128	64
SIT	2017	64	64
PRESENT-PERMS	2018	80	64

Fonte: Elaborada pelos autores.

- NOEKEON

Esse algoritmo foi criado em 2000, por Joan Daemen, Michaël Peeters, Gilles Van Assche e Vincent Rijmen (ABDUL-LATIP et al., 2010). É identificado como um algoritmo bastante resistente contra ataques conhecidos e, em comparação com outras cifras, tem o tamanho do código compacto e funciona eficientemente em várias plataformas. Tem a sua aplicação para ambiente com recursos limitados em WSN, destacando-se por ser ultra-compacto, rápido na implementação de *hardware* dedicado e por precisar de requisitos baixos de memória RAM em implementação de *software*. Utiliza bloco de 128 bits e possui tamanhos de chave de 128 bits. É mencionado por Kong et al. (2015), como sendo bastante adequado para implementações em processadores de 8 e 32 bits, adequados para RCEs. Possui um tamanho do código de 332 bytes para as sequências de criptografia e descriptografia, com 712 ciclos a uma taxa de 5,1 Mbit/s.

- CLEFIA

Foi criado pela *Sony Corporation*, em 2007. É um algoritmo de cifra de bloco altamente seguro e eficiente, que tem sua aplicação para ambientes restritivos, tais como: cartões inteligentes e dispositivos móveis, visando à autenticação e a proteção de direitos autorais. Usa bloco de 128 bits e tamanhos variáveis de chaves de 128, 192 e 256 bits (SHIRAI et al., 2007). Os resultados da implementação de *hardware* relatados são de 2.604 GE para o Type-2 CLEFIA.

- PRESENT

O PRESENT foi desenvolvido em 2007, através de uma parceria entre a *Orange Labs*, *Ruhr University Bochum* e a *Technical University of Denmark* para implementação eficiente em

hardware. É apontado por Kong *et al*. (2015) como sendo o substituto para a AES em ambientes com recursos limitados. Trabalha com bloco de 64 bits e com tamanho de chaves de 80 ou 128 bits. As suas implementações mostram-se eficientes tanto em *hardware* como em *software* e são identificadas como eficientes para as aplicações de baixa segurança. Possui *Gate Equivalents* (GE) de 1570 e menos de 2000 para RFID. O GE é uma unidade de medida padrão que representa a complexidade da tecnologia e quanto menor o seu valor, melhor o desempenho do algoritmo para dispositivos de recursos limitados.

- KLEIN

Criado em 2010, por Zheng Gong, Svetla Nikova e Yee Wei Law, foi concebido, originalmente, para WSN e etiquetas RFID, apresentando vantagens no desempenho de *software* em plataformas de sensores legados. Possui bloco de 64 bits e tamanhos variáveis de chaves de 54, 80 e 96 bits (GONG et al., 2011). Ele oferece segurança moderada e foi testado em várias situações, nas quais a maioria das cifras leves tiveram desempenhos insuficientes. Nas implementações de *hardware* para RFID, um sistema completo (incluindo os sistemas analógicos) teria um total entre 1000 e 10.000 GE. Para os circuitos de segurança, obteve cerca de 2000 GE de ocupação.

- LED - Light Encryptin Device

Desenvolvido em 2011 por Jian Guo, Thomas Peyrin, Axel Poschmann e Matt Robshaw, teve como objetivo possibilitar uma implementação mais eficiente em *hardware,* mas também razoavelmente eficiente em *software*. É considerado, segundo Kong et al. (2015), como ineficiente por consumir grande quantidade de energia por bit, comprometendo sua aplicabilidade, além de já terem sido detectadas falhas de segurança da sua aplicação. Usa bloco de 64 bits e tamanhos variáveis de chaves de 64 a 128 bits, tem sua aplicação voltada para RFID e usa 966 GE para LED64, 1040 GE para LED80, 1116 GE para LED96 e 1265 GE para LED128.

- SIMON e SPECK

Famílias de algoritmos ultraleves de cifragem de blocos desenvolvidos em 2012 pela NSA (*National Security Agency*, E.U.A), possuem aplicação para dispositivos de baixa potência, tais como RFID e dispositivos semelhantes. Usa blocos de 32, 48, 64, 96 ou 128 bits e, para cada tamanho de bloco, são suportados até 3 tamanhos de chaves, que podem ser de 64, 72, 96, 128, 144, 192 ou 256 bits (ASHUR, 2015). Implementações em *hardware* do SIMON e do SPECK com blocos de 64 bits e chaves de 96 bits precisam, respectivamente, de 838 e 984 GEs de área de circuito integrado, o que representa 35% e 41%, respectivamente, da área requerida pelo AES.

- FeW

É um *design* orientado a *software* com alta eficiência. O FeW *(A Feather-weight Block Cipher)* usa tamanho de bloco de 64 bits e tamanho de chave de 80/128 bits (chave mestra) com 32 *rounds*. A cifra leve proposta é a estrutura FeistelM, uma combinação de Feistel e estruturas generalizadas de Feistel. O cronograma da chave de FeW é similar a PRESENT (BOGDANOV et al., 2007), projeto baseado em Feistel generalizado similar ao CLEFIA (SHIRAI et al., 2007), e usa S-Box de HummingBird2 (ENGELS et al., 2011). Duas funções diferentes são usadas na função *round* e são aplicadas a duas palavras de 16 bits. A segurança é aprimorada no FeW, com correlação diferencial e

contra-ataques, sendo impossível diferenciar zero linear (KUMAR et al., 2014).

- LEA

A cifra de bloco leve LEA com ARX simples e não S-box, estrutura para palavras de 32 bits. O tamanho do bloco de LEA é de 128 bits e os tamanhos das chaves são diferentes, estes são 128/192/256-bits com 24/28/32 *rounds*, respectivamente. Operações executadas no algoritmo LEA são dois XORs chave, adição e rotação bit-wise. A função não linear usada é o módulo 232 com duas entradas de 32 bits e uma saída de 32 bits. Para difusão, a troca por palavra e rotações bit a bit são usadas. A descriptografia é semelhante ao procedimento de criptografia. O *array* de palavras de 32 bits é usado para representar a chave do LEA e o agendamento de chaves gera uma sequência de chaves de 192 bits sem misturar as palavras. As constantes são usadas para gerar chaves randômicas a partir da expressão hexadecimal da raiz quadrada de 766995, onde 76, 69 e 95 são códigos ASCII de 'L', 'E' e 'A' (SEO et al., 2015).

- ChaWskey

É um algoritmo de *Message Authentication Code* (MAC) baseado em permutação para microcontroladores de 32 bits. Chaskey é inspirado na permutação de SipHash (AUMASSON; BERNSTEIN, 2012) com 32 bits em vez de 64 bits. Ele usa a metodologia de projeto ARX (*AdditionRotation-XOR*). Operação de adição e XOR aplicada na palavra de 32 bits. O Chaskey não segue nenhum cronograma de chaves, pois a geração de chaves é feita pelo XOR com o estado e a chave *updation* envolve duas substituições e dois XORs condicionais para duas subchaves. Não requer *nonce* e, portanto, é seguro (MOUHA et al., 2014).

- HISEC

É um algoritmo de criptografia de bloco leve *Feistel*, um aprimoramento sobre PRESENT (BOGDANOV et al., 2007). O HISEC tem tamanho de bloco de 64 bits e tamanho de chave de 80 bits com um total de 15 *rounds*. As características do HISEC são as mesmas do PRESENT, sendo diferente a permutação de bit, uma vez que é aplicada em dois lados e cada lado contém 32 bits. Tem quatro camadas, com operações XOR em cada uma, todas utilizando diferentes chaves secretas. Criptografa dando não linearidade ao algoritmo usando S-box único de 4 bits em 16 *rounds*, bem como difusão por permutação de bits e a última rotação é um XOR. A cifra proposta é segura contra ataques diferenciais, integrais e bumerangues (ALDABBAGHlD et al., 2014).

- ITUBee

Os autores Karakoç et al. (2015) propuseram AKF, um novo esquema de criptografia de bloco leve Feistel com chaves alternadas chamado ITUbee, um esquema orientado a *software* baseado em AKF. O ITUbee usa o S-box do AES e reduz os requisitos de memória, consumo de energia, requisitos de tempo e é resistente a ataques de chaves relacionadas. O não uso da tecnologia de chaves alternadas em uma programação chave de estrutura Feistel torna a codificação vulnerável a ataques de chave relacionada (DINU et al., 2016). Juntamente com o esquema AKF, o ITUbee possui as chaves *whitening* nas suas camadas, S-Box de 8 bits. Para a camada de difusão, são necessárias apenas 15 operações XOR. Para fornecer resistência contra o *selfsimilarity attack* (*reflection, slide, and slidex*), diferentes permutações no lado

da criptografia são usadas por causa da adição constante de *rounds*, enquanto permutações do lado da descriptografia os *rounds* são os mesmos (KARAKOÇ et al., 2013). Constantes de 16 bits são usadas para reduzir o número de operações e evitar vazamento de informações (JEAN; NIKOLIC; PEYRIN, 2014). Existe um distintivo diferencial de chave relacionada para até oito *rounds* da cifra, utilizando uma técnica autossimilar (KARAKOÇ et al., 2015).

- LAC

É uma versão simplificada de LBlock (WU; ZHANG, 2011), *cipher block light* usando uma estrutura similar à da ALE (BOGDANOV et al., 2013). A nova cifra projetada é chamada de LAC, um *design* respeitado que usa o número de mensagem pública (PMN) como um *nonce*. A mesma chave mestra permite criptografar no máximo 240 bits. A criptografia/descriptografia é feita aceitando-se a chave mestra de tamanho 80 bits, um PMN de 64 bits, uma mensagem de cifra de texto e uma chave de autenticação de 64 bits (ZHANG et al., 2014).

- OLBCA

Banik *et al*. (2014) propuseram OLBCA, uma cifra de bloco de 64 bits com tamanho de chave de 80 bits com 22 *round*s. Cada *round* no OLBCA consiste em três camadas cada, exceto o último *round*, que tem quatro camadas. Três camadas consistem em S-box de 124 bits, permutações de bits, rotações, operação ExclusiveOR aplicada três vezes e permutação de palavras. A última camada no último *round* aplica a operação XOR na saída da terceira camada em todos os 64 bits, com 64 bits da chave sendo atualizados. O algoritmo proposto foi avaliado com três tipos de ataques: ataque diferencial, ataque integral e ataque de bumerangue. Por fim, os resultados mostraram que o OLBCA é melhor que o algoritmo PRESENT nos três fatores.

- RECTANGLE

Zhang *et al*. (2015) propuseram RECTANGLE. A cifra possui tamanho de bloco de 64 bits e um tamanho de chave de 80/128 bits. Cada *round* consiste em três operações: AddRoundkey, SubColumn e ShiftRow. A camada de substituição tem 16 S-boxes semelhante a S 4X4 em paralelo e a camada de permutação tem três rotações. Devido à sua implementação *bit-slice*, possui uma boa velocidade de *software* (ZHANG et al., 2015). Para evitar ataques de *slides* na programação da chave, são adicionadas diferentes constantes de arredondamento. A combinação de S-box e P-layer no RECTANGLE traz trilhas diferenciais/lineares limitadas. Ele fornece boa resistência contra ataques matemáticos e de canal lateral. RECTANGLE tem estrutura de matriz como AES (DAEMEN; RIJMEN, 1999), então precisa de mais ciclos computacionais (LIM; KORKISHKO, 2005).

- PICO

É uma codificação leve baseada em SPN. A cifra PICO tem tamanho de bloco de 64 bits, tamanho de chave de 128 bits e 32 *rounds*. Um grande número de S-boxes ativos é gerado relativamente em menos *rounds* para fornecer boa imunidade contra ataques lineares e diferenciais. Tem S-box forte, o que o torna robusto. O escalonamento de chaves extrai 33 sub-chaves com 64 bits a partir da chave mestra de 128 bits (SUZAKI et al., 2011). O projeto proposto faz uma fusão de S-box de um número de cifras de bloco leves e P-box de GRPs (BANSOD et al., 2015).

- LAX e SPARX

Dinu *et al.* (2016) propuseram uma família de ARX (*Modular Addition/ Bitwise Rotation/*

XOR), cifras de bloco leve de chave simétrica chamada SPARX e LAX. O LTS emprega um grande S-Box baseada em ARX chamada arx-box, juntamente com camadas lineares esparsas para adicionar não-linearidade e difusão suficiente. O ARX reduz o impacto de ataques de canal lateral ao não usar pesquisas de tabela. Essa estratégia de projeto permite implementações rápidas de *software*, minimizando as operações executadas. A cifra SPARX é projetada de acordo com a estratégia de projeto de trilha longa LTS (uma dupla de WTS (DAEMEN; RIJMEN, 2001)) e LAX completa. Há um total de 8 etapas com 3 rounds em cada etapa do Sparx-64/128, enquanto que o Sparx-128/128 usa 8 passos com 4 rounds em cada etapa e o Sparx-128/256 usa 10 etapas e 4 rounds por etapa (DINU et al., 2016).

- Lilliput com EGFN

Ali e George (2017) introduziram o algoritmo de cifra de bloco Lilliput com EGFN (*Extended Generalized Feistel Network*). A abordagem proposta implementou o método de criptografia do PRESENT (KONG et al., 2015) e um cronograma-chave semelhante ao do cronograma chave do DES. Melhorando LILLIPUT, tem um tamanho de bloco de 64 bits, chave de 80 bits, 30 rounds em que a função circular atua no nível do *nibble*. Os autores apresentam uma implementação com o método de criptografia "PRESENT"em seu algoritmo, existe uma redução de possível atraso.

- MANTIS e SKINNY

SKINNY é da família de cifras de bloco *tweakable* apresentada por Beierle et al. (2016) e projetado sob o *framework* TWEAKEY (JEAN et al., 2014), cujo objetivo é competir com o design recente da NSA SIMON, em termos de desempenho de *hardware* e *software*. O MANTIS é uma codificação de bloco *tweakable* de baixa latência com tamanho de bloco de 64 bits e tamanho de chave de 128 bits com um ajuste de 64 bits. É uma versão melhorada do MIDORI (BANIK et al., 2014), utilizando sua S-box e sua camada linear para difusão rápida.

- RoadRunnerR

Em Baysal e Sahin (2015), os autores propuseram o algoritmo RoadRunneR, o qual tem um tamanho de bloco de 64 bits e um tamanho de chave de 80 bits ou 128 bits, que exige 10 e 12 *rounds,* respectivamente. Essa técnica combina as técnicas S-box e PRIDE. Além disso, os autores propuseram a métrica ST/A que classifica as cifras através do comprimento da chave.

- DLBCA

DLBCA é um algoritmo de cifra leve de 32 bits proposto por AlDabbagh (2017). O objetivo do DLBCA é diminuir o fator de custo usando um menor número de S-boxes. Os autores avaliam o desempenho do algoritmo utilizando os ataques diferenciais e bumerangue.

- GIFT

Banik *et al*. (2017) propuseram uma cifra de bloco denominada GIFT. Os benefícios propostos para este aprimorando do PRESENT o superam em eficiência em diferentes domínios. Duas versões do GIFT são propostas, GIFT-64 com 28 *rounds* e GIFT-128 com 40 *rounds*. Ambas as versões têm um tamanho de chave de 128 bits.

- LiCi

É uma nova cifra de bloco *Feistel* leve e balanceada "LiCi", com tamanho de bloco de 64 bits, tamanho de chave de 128 bits com 31 *rounds*. Ele usa S-boxes de 4 bits, operação XOR, mudança circular de 3 vezes do lado da descriptografia, bem como mudança circular de 7 vezes do lado da criptografia. A programação

de chave inspirada no agendamento de chaves do PRESENT extrai 64 LSB da chave mestra de 128 bits e atualiza a chave mestra usando a mudança circular por 13 vezes do lado da descriptografia. Sendo a mais leve cifra entre as cifras existentes, ela precisa apenas de 1944 bytes de memória Flash e 1256 bytes de RAM (PATIL et al., 2017).

- SIT

SIT *(Secure IoT)* é um algoritmo de cifra leve proposto por Usman et al. (2017). SIT possui bloco de 64 bits, chave de 64 bits e 5 *rounds*. Esse algoritmo utiliza uma abordagem hibrida que combina estruturas Feistel e SPN, adotando operações lógicas junto com algumas trocas e substituições. A rede Feistel de funções em difusão de substituição é usada no algoritmo SIT para fusão e difusão.

- PRESENT-PERMS

Thorat *et al*. (2018) propuseram um algoritmo híbrido que combina instruções de permutação de bits rápida PERMS com box-S da cifra de bloco PRESENT. Os resultados mostram que a PERMS executa permutações arbitrárias em etapas menores que log (n), em comparação com todas as outras instruções de permutação de bit. Além disso, o PERMS tem um número menor de ciclos de CPU e GE, o que o torna mais rápido e eficiente em relação ao GRP. A técnica é implementada e avaliada em processadores ARM de 32 bits na linguagem "C".

Este estudo apresenta o grande interesse de pesquisadores no campo das cifras de blocos leves. Sendo que em CPUs de 8 bits para otimizar implementações de *hardware*, alguns projetos usam blocos de construção e, portanto, não são apropriados para implementação em *software*. Como alternativa, alguns projetos mais recentes se concentraram no desempenho da implementação de *software*.

Percebe-se que o surgimento de alguns novos algoritmos se dá pela implementação de técnicas criptográficas existentes em outras cifras, sendo essas aprimoradas e combinadas com novas técnicas.

CONCLUSÃO E TRABALHOS FUTUROS

Neste artigo, são apresentadas várias cifras de blocos leves candidatas para aplicações IoT. Os algoritmos de cifras de blocos leves são aplicações criptográficas antigas. Pode-se perceber, através deste estudo, a criticidade da segurança com relação aos dispositivos de recursos limitados, principalmente quando ligados diretamente a uma rede ou à internet. Esses dispositivos estão sujeitos a todos os riscos das redes normais, mas com poucos recursos para a implementação de estruturas de segurança. Com essa problemática, fica cada vez mais evidente que o tema necessita de muita atenção.

Essa segurança é um dos fatores críticos de sucesso dessas tecnologias no cenário atual e, pode-se perceber que existe uma grande quantidade de algoritmos disponíveis no mercado, mas não existe, ainda, uma tecnologia consolidada com relação a esses algoritmos e à definição de qual o melhor caminho a seguir.

Vários cenários ainda se apresentam, pois, essas tecnologias ainda estão em pleno desenvolvimento e novas possibilidades e riscos surgem a cada dia.

A partir deste estudo, foram identificados como possibilidades para trabalhos futuros a busca pela otimização dos códigos, visando os dispositivos de recursos limitados, uma vez que envolve a definição de um critério para escolha de algoritmos de cifras leves que possam ser otimizados, como também a implementação em ambiente de recurso restrito dos algoritmos selecionados, visando uma análise detalhada da sua forma de execução e dos

seus resultados nas simulações, com relação a desempenho, segurança, consumo de memória, processador e energia.

Esses dados podem servir de subsídio para a otimização dos códigos usados, com o objetivo de melhorar o desempenho desses algoritmos nos dispositivos de recursos limitados.

REFERÊNCIAS

ABDUL-LATIP, S. F.; REYHANITABAR, M. R.; SUSILO, W.; SEBERRY, J. *On the security of noekeon against side channel cube attacks*. In International Conference on Information Security Practice and Experience, pages 45–55. Springer.

ALDABBAGH, S. S. M. Design 32-bit lightweight block cipher algorithm (dlbca). *International Journal of Computer Applications*. 2017.

ALDABBAGH, S. S. M.; SHAIKHLI, A.; TAHA, I. F.; ALAHMAD, M. A. *Hisec: A new lightweight block cipher algorithm*. In Proceedings of the 7th International Conference on Security of Information and Networks. ACM. 2014.

ALI, M. P.; GEORGE, G. T. Optimised design of light weight block cipher lilliput with extended generalised feistal network (egfn). *International Journal of Innovative Research in Science, Engineering and Technology*. 2017.

ASHUR, T. Improved linear trails for the block cipher simon. *IACR Cryptology ePrint Archive*, 2015: 285. 2015.

AUMASSON, J.-P; BERNSTEIN, D. J. *Siphash: a fast short-input prf*. In International Conference on Cryptology in India, pages 489–508. Springer. 2012.

BAE, G. C.; SHIN, K. W. An efficient hardware implementation of lightweight block cipher algorithm clefia for iot security applications. *Journal of the Korea Institute of Information and Communication Engineering*, 20 vol. 2, 2016.

BANIK, S.; BOGDANOV, A.; ISOBE, T.; SHIBUTANI, K.; HIWATARI, H.; AKISHITA, T.; REGAZZONI, F. *Midori: a block cipher for low energy*. In International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2014.

BANIK, S.; PANDEY, S. K.; PEYRIN, T.; SASAKI, Y.; SIM, S. M.; TODO, Y. *Gift: a small present*. In International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2017.

BANSOD, G.; RAVAL, N.; PISHAROTY, N. Implementation of a new lightweight encryption design for embedded security. *IEEE Transactions on information forensics and security*, 10 vol. 1, p. 142–151, 2015.

BAYSAL, A.; SAHIN, S. *Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors*. In International Workshop on Lightweight Cryptography for Security and Privacy, p. 58–76. Springer, 2015.

BEIERLE, C.; JEAN, J.; KÖLBL, S.; LEANDER, G.; MORADI, A.; PEYRIN, T.; SASAKI, Y.; SASDRICH, P.; SIM, S. M. *The skinny family of block ciphers and its low-latency variant mantis*. In Annual International Cryptology Conference, p. 123–153. Springer, 2016.

BHARDWAJ, I.; KUMAR, A.; BANSAL, M. *A review on lightweight cryptography algorithms for data security and authentication in iots*. In Signal Processing, Computing and Control (ISPCC), 2017.

BOGDANOV, A.; KNUDSEN, L. R.; LEANDER, G.; PAAR, C.; POSCHMANN, A.; ROBSHAW, M. J.; SEURIN, Y.; VIKKELSOE, C. *Present: An ultra-lightweight block cipher*. In International Workshop on Cryptographic Hardware and Embedded Systems, p. 450 – 466. Springer, 2007.

BOGDANOV, A.; MENDEL, F.; REGAZZONI, F.; RIJMEN, V.; TISCHHAUSER, E. *Ale: Aes-based lightweight authenticated encryption.* In International Workshop on Fast Software Encryption, Springer, 2013.

BUCHMANN, J.; GÖPFERT, F.; GÜNEYSU, T.; ODER, T.; PÖPPELMANN, T. *High-performance and lightweight lattice-based public-key encryption.* In Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, ACM, 2016.

CHEN, S.; XU, H.; LIU, D.; HU, B.; WANG, H. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1, vol. 4, 2014.

DAEMEN, J.; RIJMEN, V. *Aes proposal*: Rijndael, 1999.

DAEMEN, J.; RIJMEN, V. The wide trail design strategy. In IMA International Conference on

Cryptography and Coding, Springer, 2001.

DINU, D.; PERRIN, L.; UDOVENKO, A.; VELICHKOV, V.; GROßSCHÄDL, J.; BIRYUKOV, A. *Design strategies for arx with provable bounds: Sparx and lax*. In International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2016.

DORRI, A.; KANHERE, S. S.; JURDAK, R.; GAURAVARAM, P. *Blockchain for iot security and privacy: The case study of a smart home*. In Pervasive Computing and Communications Workshops (PerCom Workshops), 2017.

ENGELS, D.; SAARINEN, M. J. O.; SCHWEITZER, P.; SMITH, E. M. *The hummingbird-2 lightweight authenticated encryption algorithm.* In International Workshop on Radio Frequency Identification: Security and Privacy Issues. Springer, 2011.

GONG, Z.; NIKOVA, S.; LAW, Y. W. *Klein: a new family of lightweight block ciphers*. In International Workshop on Radio Frequency Identification: Security and Privacy Issues. Springer, 2011.

JEAN, J.; NIKOLIC, I.; PEYRIN, T. *Tweaks and keys for block ciphers: the tweakey framework*. In International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2014.

JING, Q.; VASILAKOS, A. V.; WAN, J.; LU, J.; QIU, D. Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20 vol. 8, 2014.

KARAKOÇ, F.; DEMIRCI, H.; HARMANCI, A. Akf: A key alternating feistel scheme for lightweight cipher designs. *Information Processing Letters*, 115, vol. 2, 2015.

KARAKOÇ, F.; DEMIRCI, H.; AND HARMANCI, A. E. *Itubee: a software oriented lightweight block cipher*. In International Workshop on Lightweight Cryptography for Security and Privacy, Springer, 2013.

KITCHENHAM, B. *Procedures for performing systematic reviews*. Keele University and Empirical Software Engineering National ICT Australia Ltd, 2015.

KNEŽEVIC, M.; NIKOV, V.; ROMBOUTS, P. *Low-latency encryption–is "lightweight= light+ wait?"*. In International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2012.

KONG, J. H.; ANG, L. M.; SENG, K. P. A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. *Journal of Network and Computer Applications*, 2015.

KUMAR, M.; PAL, S. K.; PANIGRAHI, A. *Few: A lightweight block cipher.* IACR Cryptology ePrint Archive, 2014.

KUSHWAHA, P. K.; SINGH, M.; KUMAR, P. A survey on lightweight block ciphers. *International Journal of Computer Applications, 2014.*

LEE, I; LEE, K. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, vol. 4, 2015.

LIM, C. H.; KORKISHKO, T. *Mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors*. In International Workshop on Information Security Applications, Springer, 2005.

MCCANN, D.; EDER, K.; OSWALD, E. *Characterising and comparing the energy consumption of side channel attack countermeasures and lightweight cryptography on embedded devices*. In Secure Internet of Things (SIoT), 2015.

MISHRA, S. Network security protocol for constrained resource devices in internet of things. *2015 Annual IEEE India Conference (INDICON)*, 2015.

MOHD, B. J.; HAYAJNEH, T.; AND VASILAKOS, A. V. A survey on lightweight bloack ciphers for low-resource devices: Comparative study and open issues. *Journal of Network and Computer Applications*, 2015.

MOUHA, N.; MENNINK, B.; VAN HERREWEGE, A.; WATANABE, D.; PRENEEL, B.; VERBAUWHEDE, I. *Chaskey: an efficient mac algorithm for 32-bit microcontrollers*. In International Workshop on Selected Areas in Cryptography, Springer, 2014.

PATIL, J.; BANSOD, G.; KANT, K. S. *Lici: A new ultra-lightweight block cipher*. In Emerging Trends & Innovation in ICT (ICEI), International Conference on, 2017.

PAWAR, M.; AGARWAL, J. A literature survey on security issues of wsn and different types of attacks in network. *Indian Journal of Computer Science and Engineering*, 8, vol. 2, 2017.

SEHRAWAT, D.; GILL, N. S. Lightweight block ciphers for iot based applications: A review. *International Journal of Applied Engineering*

Research, 13(5):2258– 2270, 2018.

SEO, H.; LIU, Z.; CHOI, J.; PARK, T.; KIM, H. *Compact implementations of lea block cipher for low-end microprocessors.* In International Workshop on Information Security Applications, Springer, 2015.

SHIRAI, T.; SHIBUTANI, K.; AKISHITA, T.; MORIAI, S.; IWATA, T. *The 128-bit blockcipher clefia*. In International Workshop on Fast Software Encryption, Springer, 2007.

SUNDARAM, B. V.; RAMNATH, M.; PRASANTH, M.; SUNDARAM, V. *Encryption and hash-based security in internet of things*. In Signal Processing, Communication and Networking (ICSCN), 3rd International Conference on, 2015.

SUZAKI, T.; MINEMATSU, K.; MORIOKA, S.; KOBAYASHI, E. *Twine: A lightweight, versatile block cipher.* In ECRYPT Workshop on Lightweight Cryptography, 2011.

THORAT, C.; INAMDAR, V. Implementation of new hybrid lightweight block cipher. *Applied Computing and Informatics*, 2018.

USMAN, M.; AHMED, I.; ASLAM, M. I.; KHAN, S.; SHAH, U. A. *Sit: A lightweight encryption algorithm for secure internet of things*. 2017.

WU, Q. X.; HAN, L. Secure solution of trusted internet of things based on tcm. *The Journal of China Universities of Posts and Telecommunications*, 2013.

WU, W. ZHANG, L. *Lblock: a lightweight block cipher*. In International Conference on Applied Cryptography and Network Security, Springer, 2011.

ZHANG, L.; WU, W.; WANG, Y.; WU, S.; ZHANG, J. *Lac: A lightweight authenticated encryption cipher*, 2014.

ZHANG, W., BAO, Z., LIN, D., RIJMEN, V., YANG, B., AND VERBAUWHEDE, I. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58, vol. 12, 2015.