Performance Analysis of HIP Diet Exchange for WSN Security Establishment

Pin Nie
Aalto University, School of Science and Technology
Finland
pin.nie@tkk.fi

Juho Vähä-Herttua
Aalto University, School of Science and Technology
Finland
juho.vaha-herttua@tkk.fi

Tuomas Aura
Aalto University, School of Science and Technology
Finland
tuomas.aura@tkk.fi

Andrei Gurtov
University of Oulu, CWC
Finland
gurtov@cs.helsinki.fi

ABSTRACT

Wireless Sensor Nodes are powered by limited batteries and equipped with constrained processor and memory. Therefore, security protocol must be highly efficient to fit WSNs. Meanwhile, considering the large variety of WSN applications and wide deployment, scalability and interoperability are two important concerns of adopting standardized communication protocols. HIP DEX, an IETF Internet draft, provides a generic solution to establish secure connections in WSNs. In this paper, we investigate the security features of HIP DEX based on several practical attack models. We evaluate the performance efficiency of HIP DEX in terms of energy consumption and computing latency on an experimental prototype. Our empirical results show that HIP DEX is applicable for resource constrained sensor nodes to establish hop-by-hop secure connection. In order to reinforce identity protection, we also propose tentative improvements to HIP DEX. Finally, we compare HIP DEX with SSL/TLS to highlight their respective advantages in different WSN architectures.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Protocols—*Protocol verification*

General Terms

Security

Keywords

Security protocol, Elliptic Curve Cryptography, Host Identity Protocol, Wireless Sensor Networks

Q2SWinet'11, October 31–November 4, 2011, Miami, Florida, USA.
Copyright 2011 ACM 978-1-4503-0901-1/11/10 ...$10.00.

1. INTRODUCTION

Wireless sensor network (WSN) is achieving wider deployment in many applications. Security has become a crucial factor when selecting an appropriate WSN solution [1, 2], especially in safety-critical applications, such as human health monitoring [3]. Attentions to communication security and data privacy rise, due to the practical attacks [4] revealing security vulnerabilities of existing products. A number of security measures and protocols [5, 6, 7, 8] have been designed on the link layer. In contrast, the security effort on the network layer has not achieved significant progress. One major reason is that the traditional IP-based security infrastructures are too heavy to operate on tiny sensor nodes. Long lifespan and fast data acquisition allow little tradeoff when employing security features in WSNs. Lightweight and efficient cryptographic primitives are required. Consequently, Elliptic Curve Cryptography (ECC) [9] is becoming a standard security component for WSNs. Compared with the well-known RSA algorithm used in public-key cryptography (PKC), ECC achieves the same security strength with much smaller key size [10]. This feature saves considerable energy and memory on sensor nodes for security communications.

Meanwhile, the large variety of sensor nodes and wide deployment of WSN applications require a standardized security handshake protocol to guarantee interoperability in heterogeneous WSN environment. Host Identity Protocol (HIP) [11], an IETF standard, introduces a new protocol layer to establish secure signaling channel with inherent support for mobility. This protocol employs ECC [12] and has proven its feasibility on the real sensor node [13]. HIP Diet Exchange (HIP DEX) [14] is a variant of the HIP Base Exchange (HIP BEX) specifically designed for sensor devices with fewer cryptographic primitives. The goal is similar to an earlier Lightweight HIP (LHIP) proposal [15]. The difference is that instead of removing public key cryptography completely as in LHIP, only the signature is removed and the expensive Diffie-Hellman key exchange is replaced with the ECC variant better suited for sensor nodes. Moreover, LHIP relies heavily on HMAC and hash chains, HIP DEX removes cryptographic hash functions. In addition to the support of identity authentication, data encryption and message integrity, HIP DEX can also be used directly as a keying mechanism for a MAC layer security protocol in WSN radio standard, such as IEEE 802.15.4 [16].

In this paper, we investigate the security features of HIP DEX protocol in the light of practical attacks. Based on HIP DEX Internet draft, we implemented a prototype to evaluate the protocol overhead in terms of energy consumption and computing latency. Our security analysis and empirical results address potential vulnerabilities and possible bottlenecks of the current version of HIP DEX. Thus, we propose a few tentative improvements to extend the draft for better identity protection. Finally, we compare HIP DEX with another widely deployed security protocol SSL/TLS. Despite some similar cryptographic primitives and a four-way handshake protocol, these two security solutions differ from each other in their respective goals. HIP DEX aims to offer hop-by-hop protection in multihop WSNs, while SSL/TLS attempts to provide end-to-end security at the edge of WSNs between the base station and sensor nodes. Therefore, they fit different WSN architectures.

The rest of the paper is organized as follows: Section 2 elaborates the security handshake of HIP DEX protocol. Section 3 examines the countermeasures in HIP DEX against several practical attacks in WSN. Section 4 presents our implementation work and experimental results followed by proposed improvements. Section 5 makes a comparison between HIP DEX and SSL/TLS in consideration of their respective advantages in different WSN architectures. Section 6 discusses the hardware acceleration for cryptographic computations. Finally, Section 7 summarizes the paper.

2. HIP DEX PROTOCOL OVERVIEW

HIP DEX protocol consists of four messages to establish a secure direct connection between two neighbor nodes, the Initiator and the Responder respectively. These four messages are I1, R1, I2 and R2. Figure 1 illustrates HIP DEX four-way handshake protocol.

The first message, I1, includes the source host identity tag (HIT) and optional destination HIT (DST HIT). The second message, R1, contains a puzzle (a cryptographic challenge) to the initiator and also specifies the encryption algorithms supported by the responder. The third message, I2, gives the solution to the puzzle and a key wrap parameter. This message is MACed (message authentication code) to insure message integrity against tampering or corruption. The fourth message, R2, is also MACed and contains another key wrap parameter and finalizes the handshake. I2 and R2 constitute an authenticated secret key wrapped by ECDH (Elliptic Curve Diffie-Hellman) for session key generation, which will be used to encrypt subsequent data packets. Assume employing the IEEE 802.15.4 radio standard, due to the small payload space (maximum 102 bytes per frame), R1, I2 and R2 HIP messages have to consider packet fragmentation. HIP DEX protocol defines a state machine to regulate state transitions until a security association (SA) is established. Considering the possible high packet loss in WSNs, HIP DEX also specifies an aggressive transmission mechanism for I1 and I2 messages. The retransmission interval t msec depends on local policy.

HIP DEX protocol includes three important security features: identity authentication, data encryption and message integrity. Authentication of two parties is supported by the session key generated from the ECDH handshake. The optional ENCRYPTED HIP parameter also provides a password authentication within the exchange. However, the lack of digital signature implies the responder's identity cannot be verified by the initiator. Thus, R1 message is not protected and can be spoofed. Message integrity is guaranteed by Cipher-based MAC (CMAC). The benefit is that CMAC with AES is faster and has smaller memory footprint than Hash-based MAC (HMAC) [17] and sensor nodes are often equipped with hardware AES encryption chips.

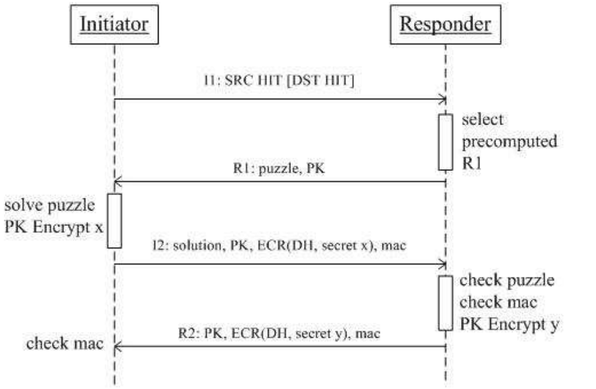

Figure 1: Sequence diagram of HIP DEX protocol

3. SECURITY ANALYSIS

In order to validate the effectiveness of HIP DEX in WSN applications, we consider five general processes. They are network initialization, data transmission, dead node resurrection, new node replenishment and invalid node revocation. At the beginning, during the WSN initialization phase, HIP DEX provides a puzzle mechanism to mitigate packet DoS attack at the responder side. This measure can protect against evil initiators exhausting the responder by flooding I1 packets in a looping script, like TCP SYN attack. Although the effectiveness of puzzle mechanism is unclear in practice [13], we find it can relax the responder in dealing with aggressive packet retransmission in the noisy environment. ECDH handshake creates symmetric encryption key for subsequent data transmissions. This step can be periodically refreshed to update the key for stronger security in the tradeoff shorter lifetime. Due to the lack of digital signature, HIT spoofing of the responder is possible. However, since HIP DEX protocol starts from the initiator, responder's HIT spoofing is a passive attack and has limited effect, depending on how often the initiator communicates the spoofing responder. Password authentication of the initiator can solve the initiator's HIT spoofing. Nevertheless, knowledge of password authentication is an optional setting and requires external configuration. In order to enhance HIT protection, we propose two tentative improvements in the next section.

In HIP DEX, session key exchange is encrypted by AES-CBC to protect from eavesdropping. Considering the better interoperability with a popular WSN open standard, Zigbee, some counter mode like AES-CTR that would not require AES decryption may be included in the future. CMAC provides an energy-efficient integrity check for small packets in WSN applications. The difference between the dead node resurrection and the new node replenishment is whether the node's identity has been authenticated and preserved or not. Dead node may replace the depleted battery and rejoin the WSN with the same identity (i.e., HIT) that was previously used. In this case, sensor nodes may reuse the key(s) to save energy, as long as it is still valid. For stronger security to prevent identity leak from the compromised node, the protocol may issue only one-time HIT for each node. No identity reuse is allowed. In node replenishment, when a new node enters the existing WSN, it acts as an initiator. HIP DEX handshake must be executed for every direct connection with the new identity. Due to the malfunction or security compromise, the WSN operator may revoke an invalid node. In this case the notification process in HIP BEX can be extended on the notify message type INVALID_HIT for revoking purpose. This message must be authenticated by every receiver with prior knowledge of the valid network controller, i.e., the base station (a.k.a. the sink node). Usually, the base station does not change, its identity and password can be hardcoded in the READ-ONLY memory of each sensor node.

We considered six practical attack models in WSN applications: radio jamming, packet DoS, replay attack, eavesdropping, spoofing/sybil attack and wormhole/man-in-the-middle attack. Spoofing and sybil attack are combined together, because they both target the identity violation. The difference is that spoofing attack misuses a valid identity, while sybil attack creates many fake identities. The man-in-the-middle (MITM) and network-level wormhole attacks are also combined, because in both cases an evil third party manipulates the communication between two nodes as if they're talking directly with each other. Since HIP DEX is a network layer protocol, it cannot protect against radio jamming at the physical and link layer. Packet DoS attacks are partly protected by the puzzle resolution, but only when attackers use equivalent devices as other sensor nodes. Otherwise, the puzzle may not generate any delay on the powerful computing device.

Replay attacks are protected by using the puzzle as a nonce and CMAC to generate keys from ECDH. Eavesdropping is well protected by the AES encryption. Spoofing/sybil attacks are partly protected by ECDH in case of non-anonymous initiators, whose public key and HIT were securely distributed beforehand. However, if one node is compromised, its identity may be used to collect more valid HITs for further attack. As elaborated in [18], sybil attack may severely subvert the quality of services (QoS) in WSN by degrading voting and fairness calculation with multiple fake identities. HIP DEX does not address this problem particularly. Due to lack of identity registration, an evil initiator could impersonate anyone it claims and broadcast numerous HITs, which may cause identity conflict. We later propose whitelist and blacklist improvements to fix this issue. HIP DEX provides password authentication as an optional security enhancement only for the responder against MITM/wormhole attacks. The initiator is still vulnerable to these attacks if a responder's identity is spoofed. Therefore, identity protection exerts significant influence to eliminate other attacks.

4. EMPIRICAL STUDY

Our experimental platform is SunSPOT rev 6 [19], a Java-based sensor node developed by the Oracle Labs. The hardware includes a 32-bit, 180MHz MCU (ARM920T) and 2.4GHz IEEE 802.15.4-compliant radio chip (TI CC2420), 512KB RAM and 4MB Flash. The sensor node is powered by a 720mAh Li-ION battery. The software provides a J2ME CLDC 1.1-compliant Java VM (Squeak) with basic operating system functionality. An optimized ECC library is available (SSL_device.jar and crypto_common.jar) and its API can be found on Java.net: Spots-security project. The advantage of a Java based sensor node is the ease of prototyping and debugging over the air. Java also gives access to large variety of existing libraries and code.

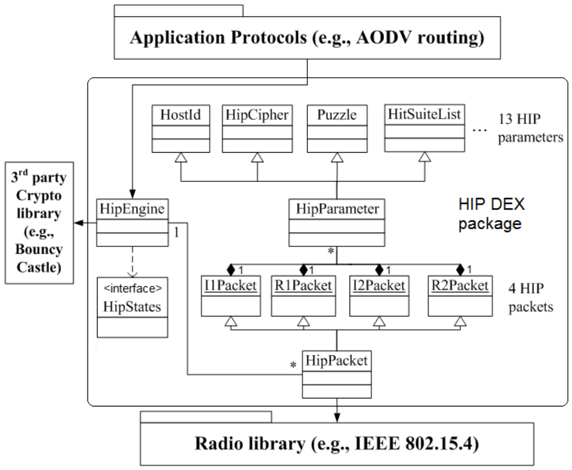

Figure 2: Modular structure of HIP DEX Java program in SunSPOT

4.1 Implementation work

According to the HIP DEX Internet draft, we implemented the handshake protocol and security primitives, including four HIP messages and thirteen HIP parameters. Figure 2 illustrates our modular program structure. Each sensor node initiates one HipEngine at the startup, which will generate the specific HipPacket based on the node's HIP state and the triggering event. To make our HIP DEX program extensible for different cryptographic libraries and radio standards, we instantiate security primitives in the HipEngine class, and the HipPacket class encapsulates HIP messages into the underlying radio packets. We do not need to consider packet fragmentation, since a simple data transport protocol has been implemented on SunSPOT supporting maximum 1260 bytes datagram.

As seen in later evaluation, the most costly operation in a HIP DEX handshake is the ECDH handshake which includes a single elliptic curve point multiplication. Therefore we need to make sure that the point multiplication code we use is optimal. The library already included a NAF (non-adjacent form) point multiplication using Jacobian projective coordinates for point doubling and addition. These are well-known optimizations used in constrained ECC environments [2]. In addition to the supported secp160r1 curve we implemented optimized modular reduction for secp192r1 and secp224r1 curves using qualities of pseudo-Mersenne primes. Consequently, we investigated the limits of common sensor node hardware regarding elliptic curve key length.

4.2 Experimental results

In order to evaluate energy consumption and computing latency of a HIP DEX handshake regarding to both initiator and responder, we divided the handshake into three parts (Puzzle resolution, ECDH handshake and CMAC calculation) and conducted stress tests on each of them. Each stress

	Energy consumption (10^{-3}mJ)	Computing latency (ms)
Puzzle generation and verification	17.95 (R)	227 (R)
Puzzle resolution	135.60 (I)	1297 (I)
ECDH handshake	143.12 (I+R)	498 (I+R)
CMAC calculation	0.44 (I+R)	4 (I+R)
Total cost	279.16 (I) 161.51 (R)	1799 (I) 729 (R)
Overhead radio	173% (I/R)	247% (I/R)

Table 1: HIP DEX overhead in terms of energy and time regarding the initiator and the responder

test iterates 100 times and calculates the average value of energy consumption and computing latency. To eliminate the side effect of debugging operations, we do not output any information in the middle of the program execution. The measurements were obtained over the air through radio transmission at the end of each test. In the following measurements, energy consumption is measured as difference of battery capacity in milliamp-hour (mAh) and computing latency is measured in millisecond (ms). To convert the difference of battery capacity (BC) mAh to energy consumption mJ, we use the following equation: VccE+3 (V) * BC/3600(mA) * time(ms). Table 1 lists the energy consumption(mJ) of all security primitives in HIP DEX, assume Vcc=3.7, K=8 and AES-128.

The overhead ratio in Table 1 is calculated by Initiator/Responder to evaluate the balance of two participants in the handshake. We tested ECDH handshake for three ECC key sizes: 160-bit, 192-bit and 224-bit. From Table 1, we can also see that the ECDH handshake costs most energy on both sides. Puzzle generation and verification have constant overhead. On initiator side of Table 1, the puzzle resolution takes significant energy and time depending on the difficulty K. It implies the puzzle mechanism, to some extent, can protect against I1 packet DoS attack. However, it also imposes considerable overhead to establish connection with new nodes. If the DoS attacker node is more powerful than the sensor node (e.g., smartphone, laptop) solving the puzzle has a very small overhead and the puzzle cannot protect the responder at all. In practice, attackers are likely using powerful devices than sensor nodes. Moreover, the puzzle resolution causes big cost on initiators during the WSN initialization phase. Table 2 lists energy consumption and computing latency regarding different parameter settings for comparison. The reference [10] gives an overview of the security strength in terms of key length in different cryptographic algorithms. At the current state of computing art (through 2010), minimum of 80 bits of security strength is required. Accordingly, ECC 160-bit key should be used to create a 128-bit AES key. Our optimized implementation demonstrates the feasibility of much higher security strength with longer ECC keys on sensor nodes. Notice that ECC 224-bit key can create a 256-bit AES key whose security lifetime extends through 2030 based on the NIST key length recommendation.

Battery lifetime analysis: according to the empirical study we assume a constant self discharge of 4% per year and 10% of the available battery energy for HIP DEX handshake. The cut-off voltage of the SunSPOT is 3.0V and the full battery provides 3.7V. As a result, only 81% of the total energy can be used. We also assume the energy consumption of security primitives takes 20% share of the total energy consumption in HIP DEX including communication. From the total cost, we can estimate that the SunSPOT can perform 256 and 148 HIP DEX handshakes regarding the actor of responder or initiator. The difference of cost between two sides of the handshake is quite large and can be mostly attributed to the puzzle resolution of the initiator. Therefore an important question is whether the puzzle mechanism is really useful in practice, considering its big cost. It might be feasible to set the puzzle complexity K to either a very low value or simply zero to solve this issue. It is worth noting that according to the current specification puzzle resolution cannot be removed completely, since the puzzle value I is also used as a nonce and initialization vector in other parts of the handshake. This issue is likely to be considered more thoroughly in later revisions.

	Energy consumption (10^{-3}mJ)	Computing latency (ms)
Puzzle resolution K=4,5,6,7,8,9,10	15.06, 24.16, 34.08, 68.41, 135.61, 221.41, 540.39	155, 245, 338, 663, 1297, 2099, 5085
ECDH handshake key=160,192,224	136.46, 208.98 301.59	498, 727 1072

Table 2: Energy consumption and computing latency with different parameter settings

4.3 Tentative improvements

As aforementioned, the current version of HIP DEX provides limited protection against DoS attack and HIT spoofing. Therefore, we propose an improvement: adding whitelist and blacklist for HIT recognition. In the most of the WSN applications, network initialization completes in short time under the supervision of the operator. Attack can be hardly launched during this stage without any notice. Hence, every node executes HIP DEX handshake to establish secure connections with neighbors within one hop distance. HITs of neighbor nodes are stored in the whitelist. After network initialization, no more unknown HITs will be accepted by any node. Whitelist is a powerful countermeasure against sybil attack. The short time frame of network initialization gives the attacker little chance to spread fake identities.

To add a new node, the trusted base station must broadcast (one-hop distance) a HIP NOTIFY message in the place where the new node is installed. This NOTIFY message type should be NEW_NODE, and it includes the HIT of the new node and the valid time period for HIP DEX handshake. After receiving this message, existing nodes will allow the new node to establish direct connections with them within the specified time. This mechanism prevents identity attacks by restricting HIT acceptance. If the trusted base station cannot be present locally, it should unicast the HIP NOTIFY message to the destination subset of the WSN. Here, we consider two cases: a cluster head (CH) in cluster-based network topology or the nodes within a specified geographic location. In cluster-based hierarchical architecture, a new node should be assigned to the nearest CH. Thus, the trusted base station unicasts the HIP NOTIFY to the CH which is closest to the new node. In the geographic routing WSN, every node is attached with a location tag either node-unique or group-unique. The trusted base station unicasts the HIP NOTIFY to the target area given by a location tag. To integrate with the geographic routing infrastructure, the new node must have the correct location tag before deployment. As aforementioned, to avoid identity leak from the compromised node, the whitelist should be updated whenever an existing neighbor node is lost or dropped out.

Meanwhile, we suggest a cross-layer security countermeasure against DoS attack by combining puzzle mechanism with the signal strength of the receiving packets (i.e., RSSI). For any receiving I1 packet with abnormally strong signal or duplicate I1 packets within short interval, the responder replies a puzzle with a large difficulty value K (e.g., >100) and put the receiving HIT into the HIT blacklist. The responder also sends a HIP NOTIFY (INVALID_HIT) to all neighbors including the bad HIT(s) in its HIT blacklist. All subsequent I1 packets with the bad HIT will be discarded. This rumor propagation mechanism guarantees fast detection and rejection of bad HITs in local cooperative manner. Our improvement proposal of HIT recognition enhances the security features of HIP DEX with little change in the protocol. Whitelist and blacklist have been widely used in many security solutions, thanks to their ease of use. Thus, we can leverage more cooperative identity protection techniques of good interoperability. Considering the constrained memory size of sensor nodes, the entry of whitelist and blacklist should be limited to one-hop direct neighbor. This strategy conforms to the HIP DEX principle of hop-by-hop security establishment.

5. COMPARISON WITH SSL/TLS

SSL/TLS is a widely adopted security handshake protocol on the Internet. Many researchers have made significant progress to migrate this protocol to WSNs [20]. In order to understand the differences and respective advantages, we make a general comparison between HIP DEX and SSL/TLS in Table 3, based on their current development status.

The criteria *Overhead* excludes the puzzle mechanism in HIP DEX, which costs considerable energy and time without practical evidence of its usefulness. However, the puzzle overhead can be resolved by reducing the puzzle complexity. The abbreviated handshake mode in SSL/TLS enables key reuse and greatly reduces time and energy costs. Similar functionality can be achieved in HIP DEX by caching the remote ECDH key and the resulting agreement. Thus, we conclude that HIP DEX is lighter than SSL/TLS. Despite the similar security features in these two protocols, the packet format and handshake of HIP DEX are easier to implement than the TLS handshake thanks to less available options and variables. This leads to a significant advantage in adopting light-weighted security solution on constrained sensor nodes. However, regarding *Identity*, SSL/TLS supports optional ECDSA-based digital signature, while HIP DEX does not contain any signing algorithm. It means that SSL/TLS has the option to verify remote identity, while HIP DEX is forced to use some kind of external whitelisting procedures, such as our tentative improvement.

Considering HIP DEX is still an IETF draft under active progress, it seems to be easier to extend HIP DEX in the near future. On the other side, SSL/TLS is a widely deployed security protocol on the Internet and is rigid to accept new changes. Moreover, HIP DEX is dedicated for WSNs and SSL/TLS has to handle different architectures on the Internet and WSNs. Hence, we give better *Extensibility* grade to HIP DEX. *Mobility* is an inherent feature supported by HIP. As a variant extension, HIP DEX inherits this advantage by nature. On the contrary, SSL/TLS must leverage some underlying protocol to support mobility, such as mobile IP. *Scalability* is a key requirement in WSNs, especially under the umbrella of the Internet of Things (IoT). Thus, any centralized entity, big chunk of data transmission and computing intensive functionality are potential bottlenecks to limit the scale of WSN applications. In HIP DEX, there is no central element required, no digital signature certificate needed and ECDH is highly efficient. HIP DEX works on the lower layer than transport layer on which SSL/TLS is built, indicating less communication overhead. Thus, we suppose HIP DEX would outperform SSL/TLS to gain better scalability in large WSN applications.

	HIP DEX	SSL/TLS
Overhead	Low (no puzzle)	Medial (no signatures)
Identity	Whitelist	ECDSA
Extensibility	High	Low
Mobility	High	Low
Scalability	High	Low
Maturity	Low	High

Table 3: Comparison between HIP DEX and SSL/TLS

One biggest difference between SSL/TLS and HIP DEX is the dependency on cryptographic hash function, like SHA-256, for key generation in addition to ECDH and AES, which might be a problem in very constrained environments where code size matters. HIP DEX only depends on ECDH and AES, making it fit into smaller space. This difference could be fixed in SSL/TLS by defining a CMAC based pseudo-random function as done in HIP DEX and not using the signing capabilities of SSL/TLS, making their cryptographic requirements the same. Such modification is not currently available, but would be worth considering in the future. It is also worth noting that some external whitelisting of identity as described in the section 4.3 is necessary to amend the current version of HIP DEX. So that filtering technique can be employed to configure trusted set of devices at runtime.

Maturity is an important concern when adopting new security protocol in WSN applications. In order to seamlessly connect with the existing infrastructure (e.g., Internet, mobile network), wide deployment is preferred on both sides. SSL/TLS is a standard component on the Internet via HTTPS over browsers and adopted by many open source projects. HIP is a relatively new protocol and has not achieved considerable deployment in practice yet. Thus, SSL/TLS is more mature than HIP DEX. By comparison, we conclude that currently HIP DEX is better suitable for large WSN application scenarios where peer-to-peer communication and mesh networking are prominent. On the other hand, SSL/TLS with signatures and stronger cryptographic primitives fits small WSNs equipped with powerful sensor nodes and a central sink node in a star network topology, typically in WPAN. Furthermore, the digital signing algorithm is mandatory in safety-critical applications and privacy-sensitive data collection, such as healthcare.

6. DISCUSSION

We notice that the hardware acceleration module offers significant performance improvement for cryptographic operations. For example, the multiply-accumulate unit in MSP-430 MCU can be used to speed up the multiplication of long integers, a core operation in for ECC computations. As a result, the computing latency of the ECDH handshake on the lower class sensor node [20], Telos Motes, is similar to our experimental results on the more powerful platform SunSPOT. Furthermore, thanks to the low class hardware, Telos Motes consumes less energy to complete the ECDH handshake with the same security parameters.

The idea of reducing cryptographic primitives to the bare minimum of ECDH key exchange and AES encryption offers efficient security solution for constrained sensor nodes and should be considered on other platforms as well, such as smartphones. It may not only save the processing time and energy drastically, but it also does reduce the big code footprint of cryptographic hashes and take advantage of the existing AES hardware acceleration.

7. CONCLUSION

In this paper, we analyzed security features of HIP DEX protocol, an IETF Internet draft, and also evaluated its overhead in terms of energy consumption and computing latency on the real device. Our prototype and empirical results prove the feasibility of this new security solution on WSNs. We also proposed improvements to enhance identity protection in HIP DEX draft. By comparing with SSL/TLS, we gained better understanding to take advantage of both solutions in their respective application scenarios. Finally, we discuss the hardware acceleration for better performance of cryptographic operations on sensor nodes. Based on our analysis, it is worth to consider bootstrapping security association using HIP DEX for WSNs in the future.

8. REFERENCES

[1] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, 2004.

[2] R. Roman, C. Alcaraz, and J. Lopez, "A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes," *Mobile Networks and Applications*, vol. 12, 2007.

[3] A. Triantafyllidis, V. Koutkias, I. Chouvarda, and N. Maglaveras, "An open and reconfigurable wireless sensor network for pervasive health monitoring," in *PervasiveHealth'08: Proc. of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, 2008.

[4] D. Halperin, T. Heydt-Benjamin, B. Ransford, S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *IEEE SP'08: Proc. of the 2008 Symposium on Security and Privacy*, 2008.

[5] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys'04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004.

[6] Y. Xiao, H.-H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, 2006.

[7] L. Lighfoot, J. Ren, and T. Li, "An energy efficient link-layer security protocol for wireless sensor networks," in *IEEE International Conference on Electro/Information Technology*, 2007.

[8] V. Misic, J. Fang, and J. Misic, "Mac layer security of 802.15.4-compliant networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 2005.

[9] D. R. L. Brown, "Standards for efficient cryptography: Elliptic curve cryptography," SECG, Tech. Rep., 2009.

[10] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key managementĹcpart 1: General(revised)," NIST, Tech. Rep., 2011.

[11] R. Moskowitz, T. Heer, P. Jokela, and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)," IETF, Tech. Rep., 2011.

[12] O. Ponomarev, A. Khurri, and A. Gurtov, "Elliptic curve cryptography (ecc) for host identity protocol (hip)," in *ICN'2010: Proc. of the 9th International Conference on Networks*, 2010.

[13] A. Khurri, D. Kuptsov, and A. Gurtov, "On application of host identity protocol in wireless sensor networks," in *IEEE MASS'2010: The 7th International Conference on Mobile Adhoc and Sensor Systems*, 2010.

[14] R. Moskowitz, "HIP Diet EXchange (DEX)," IETF, Tech. Rep., 2011.

[15] T. Heer, "LHIP - Lightweight Authentication for the Host Identity Protocol," Master's thesis, University of Tübingen, August 2006.

[16] "IEEE 802.15.4-2006 Radio Specification for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE 802.15 working group, Tech. Rep., 2006.

[17] T. S. Denis, *Cryptography for Developers*. Syngress Publishing, 2006.

[18] N. James, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *IPSN'04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004.

[19] "Sun small programmable object technology (sun spot) theory of operation," Sun Labs, Tech. Rep., 2007.

[20] V. Gupta and M. Wurm, "The energy cost of ssl in deeply embedded systems," Sun Labs, Tech. Rep., 2008.