Path Metric Authentication for Low-Power and Lossy Networks

Lin Ye, Viktoria Fodor, Thanassis Giannetsos and Panos Papadimitratos
ACCESS Linnaeus Center, KTH, Royal Institute of Technology,
Stockholm, Sweden

ABSTRACT

Cyber physical systems often require sensor networks to perform unattended operation for a long time, while providing performance guarantees for monitoring and control applications. Since this poses requirements on the formed wireless sensor network topology, routing protocols provide a means to construct topologies according to complex objective functions, involving several routing metrics. As the metric values affect the emerging topology significantly, they need to be protected, to avoid topology formation attacks from malicious users. In this paper we consider the specific case of RPL based routing and propose a lightweight authentication approach to protect RPL path metrics. We evaluate the cost of metric authentication and show how to trade-off the introduced tree construction delay and the achieved metric accuracy.

1. INTRODUCTION

Low-power wireless networks play a key role in the construction of cyber physical systems. They are often required to cover a large geographic area in harsh radio environment, need to work unattended utilizing limited energy resources, and have to support a variety of network applications, from monitoring to distributed data processing and control. To satisfy all these requirements, the topology of these low-power and lossy networks (LLNs) and the applied routing has to fulfil multiple objectives, for example, ensure end to end delays as well energy consumption balancing.

IETF proposed the Routing Protocol for Low-Power and Lossy Networks (RPL) [1] with the objective of defining a routing solution that fits well these networks. To allow diversity in network performance requirements, RPL allows the use of user defined routing metrics and routing objective functions to control the topology of the emerging routing tree. Specifically, IETF proposes the expected transmission count (ETX), energy, throughput and delay as suitable metrics [2]. Objective functions with combination of several metrics are introduced to trade-off delay, reliability or network lifetime in [3][4][5].

CySWater'15 , April 13-16, 2015, Seattle, WA, USA

http://dx.doi.org/10.1145/2738935.2738941

Routing metrics that describe the quality of a transmission path, that is, path based metrics, are transmitted as a part of the topology construction messages. Falsification of these metrics can seriously distort the network topology, leading to sub-optimal routes and even easy implementation of sinkhole attacks [6]. While RPL can utilize link layer security primitives [7] to encrypt frame information, these do not protect against insider attackers claiming false metrics.

A variety of solutions is described in the literature to protect routing metrics in general multihop wireless networks [8][9] , but they are not applicable for our scenario, as they require bi-directional paths, or result in a large message size. Recent works that address routing security threats specific to RPL concentrate on the rank parameter of RPL. The objective of VeRA [10] is to provide efficient secure tree reconstruction with global repair, and proposes low cost authentication of the version number, that identifies the new routing tree, and the rank, that gives the distance of the node from the root. It uses hash chains and message authentication code (MAC). VeRA plus parent fail over [11] targets sinkhole attacks, and proposes to add an unheard node set field in the routing control messages from the root, such that nodes on the list can detect malicious parents. TRAIL [12] aims at avoiding rank falsification by initiating a positive rank attestation of suspected parent nodes.

In this paper we propose to complement the above RPL authentication mechanisms by protecting even the routing metric values in the RPL route construction messages, as, under complex routing objectives they determine the position of the nodes joining the tree. We suggest to use the effective cryptographic primitive of one way hash chain to authenticate path metrics which always increase or decrease. We demonstrate how to trade-off the resource requirements of the security mechanism with the accuracy of the established network topology. We show that the proposed solution meets the requirements of LLNs, such as simple key prerequisite, support for all path metrics [2][13], full integration with the RPL route construction mechanisms, and controllable accuracy and overhead.

The rest of the paper is organized as follows. Section 2 gives the requirements towards the path metric authentication. Section 3 briefly describes RPL tree construction mechanisms. The proposed security approach is presented in Section 4, including the cryptographic technique, the path metric representation, and the RPL integration. In Section 5 we evaluate the performance of the metric protection. Sec-

tion 6 concludes our work.

2. SYSTEM ASSUMPTIONS AND REQUIREMENTS

We consider LLNs using RPL to construct the routing trees. We assume that the root of the network is trusted, has a private key, and the computational capability to sign messages. Nodes that may want to join the network know the public key of the root. The public key is distributed at the installation of the nodes, or by an extra public key distribution infrastructure. We do not assume trusted hardware at the nodes.

The requirements towards the path metric protection solution are the following:

1) Protect against the falsification of path based routing metrics, specifically, avoid that malicious users claim better metric and attract significant network traffic this way;

2) Applicable for protecting the variety of path-based routing metrics proposed;

3) Lead to little communication overhead, and comply the link layer protocols typically used in LLNs;

4) Do not lead to significant computational delays in the resource limited sensor nodes.

As the network environments and the requirements of the applications may vary significantly, it is an advantage if the proposed solution can be tuned accordingly.

3. ROUTING IN LLNS

RPL is designed by IETF ROLL working group to meet the core requirements of LLNs, addresses scalability, loss resilience and control overhead minimization. To provide resilience, RPL constructs a tree-like topology, Destination-Oriented Directed Acyclic Graph (DODAG), allowing multiple path towards a root node. The same physical infrastructure can accommodate DODAG *instances* serving different applications. There are three key RPL routing messages: DIO, DAO and DIS, all carried as layer three IP ICMP messages. DIO is used to propagate information for DODAG construction, DAO is optionally used to setup the reverse path, while nodes use DIS to join to an existing DODAG.

The DIO header contains the DODAG instance, ID and version numbers, set by the root (we will refer to these fields as i, d, v). It contains the node rank r, which is a mutable field, set by the node transmitting the DIO message, and reflects the position of the node in the network topology, relative to the root. The DIO contains also optional fields to transmit metric data, the quantified value of the state of the node, the link, or the route to the root, like the expected transmission count (ETX), latency, throughput or available energy.

RPL topology formulation starts from the root. To construct the topology, the nodes need to be aware of some configuration information, related to the new DODAG instance or version, like the objective function to be used, maximum rank, the metric types and parameters, [13], or security related information [10]. This information is preferably propagated in a bootstrap phase, as most of this information does not need to be updated for each new tree reconstruction. After bootstrap, the root initiates the DIO, containing $i.d.v.r$, and distributes the DIO to its neighboring nodes. Receiving a DIO message, intermediate nodes select the set of preferred parents based on the received metrics, and applying an objective function $OF(\cdot)$. Then they set a rank that is larger than that of the parents, update the metric and rank fields of the DIO message, and propagate it further in the network.

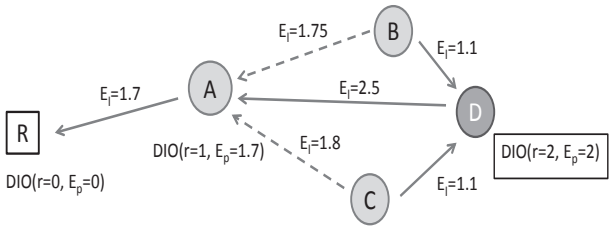


Figure 1: Node D claims false path metric E_p, so that nodes B and C select it as parent.

Figure 1 shows that by claiming false path metric E_p, a node, node D on the figure, can distort the tree topology, even if the rank values are protected. Nodes B and C select D as parent, while the optimal parent would be node A, with path ETX $E_p = 3.45$ and 3.5 respectively.

RPL provides topology maintenance to ensure functioning routing despite the lossy, unreliable environment. Global repair is initiated by root, and leads to the fundamental reconstruction of the DODAG, and includes the update of the version number, and optionally the update of some configuration information [14]. Local repair is initiated by the network nodes and requires local DIS and DIO message exchange, similar to the case when a new node joins. As shown in [15], local repair significantly outperforms global repair in large networks, leading to lower maintenance overhead and lower connectionless time. Therefore, in this paper we propose a metric authentication solution that aims at efficient initial DODAG construction and maintenance with local repair.

4. ROUTING METRIC PROTECTION

We propose to protect the RPL routing metrics in the DIO by applying one way hash chains. In this section we describe the construction of the chain, the representation of the metrics, and the integration of the hash chain based metric authentication in the RPL routing procedures. Specifically, the objective of the metric protection is to ensure that nodes can not claim metrics better than the actual ones and initiate sinkhole attacks this way. Notation used throughout the paper is summarized in Table 1.

4.1 One Way Hash Chain

Protection against the falsification of data is known as message authentication. The authentication of a message requires to verify that the content of the message is not altered, and the source is authentic. It can be performed by several approaches: conventional symmetric encryption, public key encryption (signature), message authentication code (MAC), and the one way hash function [16]. The first three approaches are not appealing in resource constraint LLNs for hop by hop changed mutable messages, as they are either computationally heavy or depend on expensive key management. One way hash authentication, introduced

Table 1: Summary of notations

Notation	Definition
$OF(\cdot)$	Objective function
$H(\cdot)$	Hash function
m	Hash chain length. e.g. 100
x, y	Metric value
$\|x\|$	Hash value
$v_{\|x\|}$	$v_{\|x\|}$=$H^{m-\|x\|}()$, Cryptographic digest of x
r	Rank value
n	Network depth. e.g. 3, 5, 10
$E_{l,i}$	Link ETX of i^{th} node along the path
$E_{p,i}$	Path ETX reported by the i^{th} node
E_{maxl}	Maximum link ETX value. e.g. 10
E_{maxp}	Maximum acceptable path ETX
τ	Computation time of one $H(\cdot)$
t_0	Computation time for ETX value 1
E_{accu}	E_{accu}= E_{maxp} /m, The smallest difference in the ETX chain

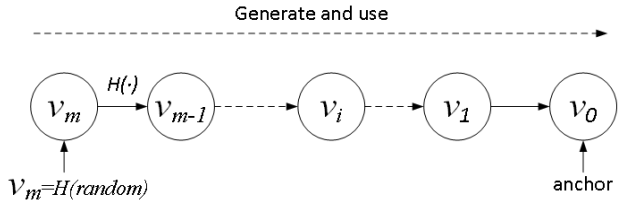


Figure 2: One way hash chain

in [17], can be an attractive solution for LLNs, as it is symmetric cryptography and allows relatively fast computation.

Figure 2 illustrates the generation and the use of the one way hash chain. To generate a hash chain of length m, the sender selects a random number and repeatedly applies a hash function, such that $v_m = H(random)$, and $v_i = H(v_{i+1})$. The last value generated, v_0, is called *anchor*. The source first safely publishes the anchor as a commitment to all possible receivers, and it is used in the followings to verify the chain elements. Then, any transmitted message i is authenticated as follows. The sender sends (i, v_i), that is, the message i together with its cryptographic digest. The receiver authenticates message i by evaluating $H^i(v_i)$==v_0, that is, again applying the hash function i times.

We use the above one way hash chain to protect the mutable, decreasing path based routing metrics. We utilize the preimage resistance of the hash chain (that is, it is computationally prohibitive to generate v_i, knowing v_{i-1}) to ensure, that nodes can not claim metric values higher than the ones received from their parents.

4.2 Path Metric Representation

One way hash chain based authentication described in this paper requires the mutable path metric values to be decreasing along the path, and they need to be represented by an integer value for the hash operation.

As proved in [18], path metrics need to be monotonic along the paths, to ensure consistent, optimal and loop-free routing. This is fulfilled, if metrics are aggregated based on addition (ETX or end-to-end delay are prominent examples), min-max operation (like path energy or throughput), or via multiplication (for example reliability, or packet reception probability)[9].

Let us now address the integer value representation. We introduce the term *metric value* x, for the actual value of the metric. Often a maximum $x_{\max}$ is given by the protocol implementation. *Objective value* $\bar{x}$ gives the binary number that represents the metric in the DIO. The range of the objective value is defined by [2], allocating 8-16 bits for the different metrics. As hash chain length of 2^{16} would lead to unacceptable authentication delays, we introduce m as the length of the hash chain, and quantize $\bar{x}$ to m steps, resulting in the *hash value* $|x|$. Moreover, for naturally increasing path metric values, like ETX, $|x|$ needs to be transformed to a decreasing one, by applying $|x| \leftarrow m - |x|$.

Clearly, m affects the smallest difference that can be represented by the authentication, therefore, we define the ratio of the highest acceptable metric value $x_{\max}$ and m as the accuracy of the authentication. As large m also means long hashing operations, the accuracy and authentication delay tradeoff needs to be evaluated. As metrics differ in the span of possible values, the quantization needs to be metric specific. We look at energy and ETX based metrics as examples.

Energy based routing metric for RPL is discussed in detail in [4]. It suggests to use the minimum node energy level on the path to the root as metric, as it reflects the lifetime of the path, and follows the IETF specification [2], representing the node's residual energy (and consequently even the path energy) on a scale of 2^8, utilizing 8 bits of the DIO. This path metric is clearly non-increasing, with a maximum of $x_{\max} = 100$, and therefore we need to address its quantization only.

Path ETX is proposed and evaluated in [2][19]. It is additive over the concatenated links, and protocol realizations give a reasonable upper value at around ETX=100 [20]. ETX is proposed to be represented in 16 bits in the DIO, by having ETX$\cdot 2^7$ as objective value. To avoid performing hashing $2^8 \sim 2^{16}$ times, we propose to consider the objective value as a fixed point representation, where only the integer part is protected by authentication. The position of the fixed point gives the required hash chain length, $m = 2^k$, if k bits are considered for the integer value. The fractional part gives space to the attacker to declare false path metric. The fixed point therefore needs to be positioned according to the required authentication delay and metric accuracy trade-off.

4.3 RPL Metric Protection Procedures

The RPL metric authentication procedure contains a bootstrapping phase, the topology formation, and finally the maintenance of the topology. It is assumed that the root has a private key and all the nodes joining the network know the public key.

To bootstrap the network, the root, as the network topology constructor, initiates the list of metrics to be considered, specifies the metric name, scope $x_{\min}, x_{\max}$, the length of the hash chain m, and sets the one bit decrease indicator $decInd$. If $decInd = 0$, then the metric value is increasing on the path, and a decreasing hash value needs to be enforced as described in Section 4.2. Last but not the least, the root specifies v_0, the anchor of the hash chain. Root prepares a pre-agreed message msg conveying metric parameters and other system configuration information, and signs it using its private key. Then it disseminates the message together with $Sig(msg)$ to the whole network. Nodes receive the

Table 2: Messages with metric protection, for any node A in the network, and a node B joining later.

Bootstrapping		
root	:	msg=((metricName, $[x_{\min}, x_{\max}]$,m,decInd,v_0), maxGlobalRepairRate)
$root \rightarrow *$	:	(msg,Sig(msg))
Topology setup		
$root \rightarrow *$	:	DIO(x_r, $v_{\|x_r\|}$, $i.d.v.r_{root}$)
$A \rightarrow *$	:	DIO(x_A, $v_{\|x_A\|}$, $i.d.v.r_A$)
New node B joins		
$B \rightarrow *$	:	DIS
$A \rightarrow B$	:	DIO(msg, Sig(msg),x_A, $v_{\|x_A\|}$, $i.d.v.r_E$)

configuration message, verify it using the public key of the root, save it and then propagate it further. (See Table 2 and Algorithm 1 for details.)

After bootstrapping, the root then initiates the topology formation, by sending a DIO$(x, v_{|x|}, i.d.v.r)$ including the metric x and its one way hash chain digest $v_{|x|}$. Each member node, upon receiving a DIO, verifies that it is from a trusted chain, by performing $|x|$ hashing functions and comparing the result to the chain anchor v_0. The node then selects the parent node according to some objective function OF, and sets its rank. It generates the new path metric y, and derives the new hash digest. If the path metric value was decreased from x to y, then $|x| - |y|$ hash operations needs to be performed to generate the new digest. Finally, the node prepares and transmits the new DIO $(y, v_{|y|}, i.d.v.r)$ (See Algorithm 2 for details).

During the lifetime of a the DODAG version, topology maintenance is performed for local repair and to accommodate joining nodes. At local repair the node detaches first, then the same procedure is followed. The joining node sends a DIS. The responded DIO now needs to convey msg and $Sig(msg)$, as the joining node may not have received the bootstrap message. The verification of msg is effective, because in most local repair case, a node only needs to compare its local stored msg and $Sig(msg)$ with the received one, instead of digital signature verification.

Algorithm 1: Topology setup procedure with metric authentication at the root.

```
if bootstrapping;
then
    v_m = H(random);
    for i ← m to 1 do
        v_{i-1} = H(v_i);
    send(msg,Sig(msg));
if topology setup;
then
    sendDIO(x,v_|x|, i.d.v.r_root);
```

Algorithm 2: Topology setup with metric authentication at the nodes.

```
if bootstrapping;
then
    receiveAndPropergate(msg,Sig(msg));
if topology setup;
then
    receive(x, v_|x|, i.d.v.r);
    if v_0 == H^|x|(v_|x|);
    then
        r = OF(x, r_received);
        v_|y| = H^(|x|-|y|)(v_|x|);
        sendDIO(y, v_|y|, i.d.v.r);
    return;
```

5. EVALUATION

5.1 Security Analysis

Let us consider possible ways of metric falsification. First, an attacker may want to claim significantly better metric than the actual one. However, to claim a metric value better than the one received from the parent node, the malicious node should be able to generate $v_{|x|+i}$, such that $H^i(v_{|x|+i})$ equals $v_{|x|}$, which is not possible due to the preimage collision resistance of the one way hash chain. Second, the attacker can claim slightly better metric than the one of the parent, with a difference within chain accuracy. Therefore, the hash chain length m needs to be selected to find an acceptable accuracy, that, under attack, does not lead to significantly sub-optimal topology construction. With the possibility of tuning the chain length, the proposed solution satisfies Requirement 1 in Section 2.

As path metrics are naturally monotonic along the path, hash chain based verification is suitable for them, fulfilling Requirement 2.

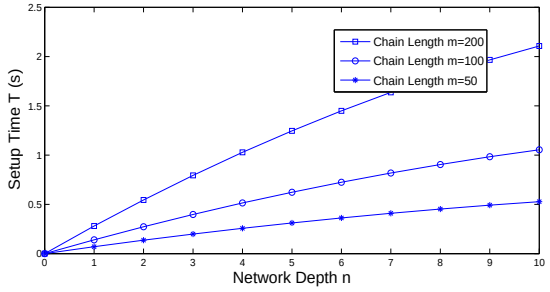
The metric authentication solution, proposed in this paper, is not appropriate for all scenarios. Specifically, stronger protection is needed, if the attacker can get significant gain by claiming i) the same metric as its parent, ii) a metric that is poorer than the one of the parent but still better than the actual one, or iii) extremely poor metric. Moreover, the solution may be expensive under frequent global repairs, due to the need of the time-consuming digital verification of the chain anchor.

5.2 Performance Evaluation

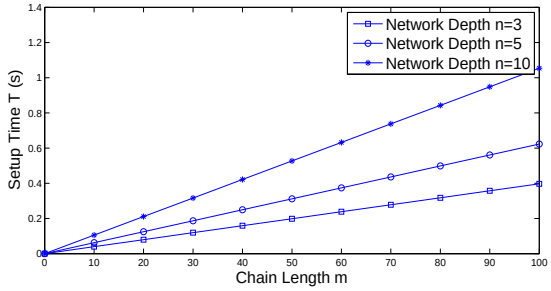
5.2.1 Communication Overhead

The communication overhead of the message authentication is low, if the number of additional packets to be transmitted in the network as well as the size of these packets is limited. Therefore, the objective of the design of the authentication scheme is to convey security related information within packets that are anyway transmitted during the topology setup procedure. In LLNs the link layer frame sizes are typically limited by the IEEE 802.15.4 standard of 127 Bytes. As the frame may contain 25 Bytes header, 21 Bytes link security overhead and 2 Bytes compressed IPv6 header, it leaves 79 Bytes for IP payload, carrying the routing control messages [21].

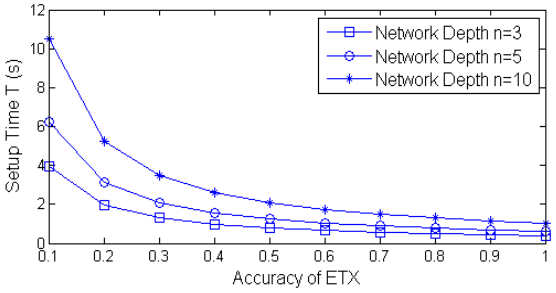
Considering the bootstrap message, the typically 20 Bytes chain anchor is carried together with existing legacy configuration information and the signature. The signature size depends on the total message size and is in the range of 30 to 40 Bytes [22]. Consequently, to accommodate security



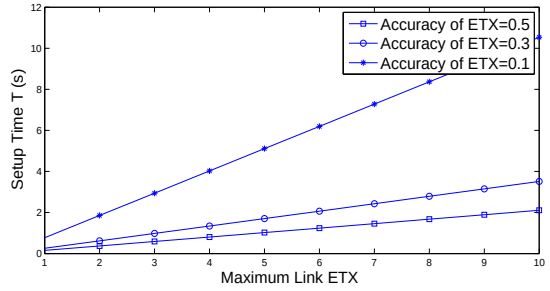
(a)



(b)



(c)



(d)

Figure 3: The additional network setup delay introduced by the metric authentication procedure, as a function of (a) the network depth, (b) the hash chain length, (c) the metric accuracy (E_{maxp}=100, E_{maxl}=10, τ=1.4ms) and (d) the link ETX span($n_{max} = n = 10$).

Table 3: Delay cost of metrics authentication

	root	member node
Generate signature	$ECC_{gen} = 1.9s$	N/A
Verify signature	N/A	$ECC_{ver} = 2.4s$
Generate hash chain	O(n)·SHA	N/A
Verify hash item	N/A	O(n)· SHA

related information, the bootstrapping needs one additional frame to be transmitted.

Without the optional metrics, the DIO size is 12 Bytes. Each of the metrics adds 1-2 Bytes, and a 20 Bytes digest. Therefore, the 79 Bytes IP payload can carry the complete DIO, as far as the number of metrics is limited to 2-3, which is rather typical in proposed routing schemes [3][4][5]. Consequently, the proposed scheme fulfills Requirement 3 on limited communication overhead.

5.2.2 Computation Overhead

Finally, we evaluate Requirement 4, that is, the computational overhead of the metric authentication. For computational delays of generating and verifying signatures and for performing hash operations we consider the values measured for MICAz sensors, as given in Table 3 [23]. The signature verification delay is high at each node, and the end to end delay increases with the number of nodes traversed. As signatures are not propagated and verified frequently, we focus on the overhead of the one way hash chain based metric authentication, considering ETX as example.

Let $E_{l,i}$ and $E_{p,i}$ be the link and path ETX values at node i, E_{maxp} be the maximum acceptable path ETX value, E_{maxl} be the maximum acceptable link ETX value, m the length of the hash chain, and n the longest path length in the resulting topology. To ensure that E_{maxp} accommodates the the longest possible path, it needs to be set such as $n \leq E_{maxp}/E_{maxl}$. We denote by τ the time needed for one hash operation $H(\cdot)$. We can then express the end to end delay of metric authentication and hash digest generation as:

$$\begin{aligned} T &= \sum_{i=1}^{n} T_i \\ &= \sum_{i=1}^{n} [(E_{maxp} - E_{p,i}) \cdot t_0 + E_{l,i} \cdot t_0] \\ &= \sum_{i=1}^{n} [(E_{maxp} - \sum_{j=i}^{i} E_{l,j}) + E_{l,i})] \cdot t_0. \end{aligned} \quad (1)$$

Where T_i is the delay at the i^{th} node along the path. It consists of (E_{maxp} - $E_{p,i}$)t_0, delay for verification of the received hash digest and $E_{l,i}t_0$ delay for the generation of the new hash digest. Time $t_0 = \tau \cdot$ m/ E_{maxp}, and reflects the quantization of the metric values.

Let us consider a uniform link ETX, $E_l \sim U[1, E_{maxl}]$, $\overline{E_l} = \frac{1+E_{maxl}}{2}$. This gives us the average end to end delay:

$$\overline{T} = [1 - \frac{(n-1)(E_{maxl}+1)}{4E_{maxp}}]mn\tau. \quad (2)$$

Clearly, $\overline{T}$ increases not only with n, but also with the tunable hash chain length parameter m. Figures 3.a and 3.b show the delay as a function of n and m respectively, consid-

ering τ=1.4ms as measured on MICAz nodes and $E_{maxl} = 10$ and $E_{maxp} = 100$. The results show that the delay increases slower than linearly with n and linearly with m, and both of these parameters have significant effect. To avoid large delays, m needs to be limited, especially in large networks. Therefore, we evaluate also the effect of limiting m, measured in the accuracy of the metric protection. Specifically, the path ETX accuracy that can be ensured is $E_{accu} = \frac{E_{maxp}}{m}$. Substituting m in (2), we can evaluate the delay and accuracy tradeoff, as shown on Figure 3.c. The results show that high accuracy requires long authentication delays. However, the delay decreases fast with decreased accuracy, and therefore we can conclude that the proposed solution can efficiently control the authentication delay and metric accuracy tradeoff.

Finally, with Figure 3.d we evaluate the effect of the uncertainty of the link quality, by changing the E_{maxl} value, while keeping the maximum path length and the required accuracy fixed. The introduced delay increases linearly with E_{maxl}, and the gradient increases with E_{accu}, showing that the metric authentication delay can be minimized, if a-priory information on the expected link quality and on the accuracy requirement is available.

6. CONCLUSION

In this paper, we proposed and analyzed a one way hash chain based approach to protect monotonic path metrics in RPL for LLNs. The preimage resistance of the hash operation prevents the attacker from claiming better metrics than the one advertised by its parent, and thus it can not distort the network topology to attract data traffic.

We showed that the hash chain based solution suits practical LLN deployment, as it does not depend on complex key management schemes, allows fast computation and does not introduce significant communication overhead and is is applicable for typical RPL path metrics and metric aggregation methods. As the delay of the hash chain based authentication depends on the length of the chain, we described the quantization process needed for the authentication. Considering the popular ETX metric, we showed how the introduced delay depends on the chain length, on the network size, and on the link quality metric range, and demonstrated, that by tuning the length of the chain, the delay can be effectively traded off by some the allowed inaccuracy of the metric authentication.

7. REFERENCES

[1] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *http://www.ietf.org/rfc/rfc6550.txt*, 2012.

[2] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," *http://tools.ietf.org/html/rfc6551*, 2012.

[3] P. Di Marco, C. Fischione, G. Athanasiou, and P.-V. Mekikis, "Harmonizing MAC and Routing in Low-Power and Lossy Networks," in *IEEE GLOBECOM*, Dec 2013.

[4] P. O. Kamgueu, E. Nataf, T. D. Ndie, and O. Festor, "Energy-Based Routing Metric for RPL," *Research Report RR-8208, INRIA*, 2013.

[5] X. Yang, J. Guo, P. Orlik, K. Parsons, and K. Ishibashi, "Stability Metric Based Routing Protocol for Low-Power and Lossy Networks," *IEEE ICC*, 2014.

[6] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things," *International Journal of Distributed Sensor Networks*, 2013.

[7] K. Krentz, H. Rafiee, and C. Meinel, "6LoWPAN Security: Adding Compromise Resilience to the 802.15.4 Security Sublayer," *ACM Workshop on Adaptive Security*, 2013.

[8] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Ad Hoc Networks*, 2003.

[9] P. Papadimitratos and Z. Haas, "Secure Route Discovery for QoS-Aware Routing in Ad Hoc Networks," *IEEE Sarnoff Symposium*, 2005.

[10] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-Version Number and Rank Authentication in RPL," *IEEE MASS*, 2011.

[11] K. Weekly and K. Pister, "Evaluating Sinkhole Defense Techniques in RPL Networks," *IEEE ICNP*, 2012.

[12] H. Perrey, M. Landsmann, O. Ugus, T. Schmidt, and M. WÃd'hlisch, "TRAIL: Topology Authentication in RPL," *arXiv preprint arXiv:1312.0984*, 2013.

[13] N. Leligou and T. Zahariadis, "Trust-Aware Routing Protocol Specifications," *EU FP7 VITRO, D4.2*, 2011.

[14] J. W. Hui, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams," *http://www.ietf.org/rfc/rfc6553.txt*, 2012.

[15] J. Tripathi, J. Oliveira, and J. Vasseur, "A Performance Evaluation Study of RPL: Routing Protocol for Low-Power and Lossy Networks," *IEEE CISS*, 2010.

[16] W. Stallings, "Network Security Essentials: Applications and Standards," *Pearson Education India*, 2007.

[17] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, 1981.

[18] J. L. Sobrinho, "Network Routing with Path Vector Protocols: Theory and Applications," *ACM SIGCOMM*, 2003.

[19] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *ACM Wireless Networks*, 2005.

[20] SICS, "ContikiRPL," *http://contiki.sourceforge.net/docs/*, 2010.

[21] N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *http://tools.ietf.org/html/rfc4944*, 2007.

[22] I. F. Blake, G. Seroussi, and N. Smart, "Elliptic Curves in Cryptography," *Cambridge University Press*, 1999.

[23] "Micaz," *openautomation.net/uploadsproductos/micaz_datasheet.pdf*, 2007.