RFID Tags Authentication by Unique Hash Sequence Detection

Abdallah Y. Alma'aitah
Electrical and Computer Engineering Department
Queen's University
Kingston, Ontario, Canada
Email:8aa14@queensu.ca

Hossam S. Hassanein
School of Computing
Queen's University
Kingston, Ontario, Canada
Email:hossam@cs.queensu.ca

Mohamed Ibnkahla
Electrical and Computer Engineering Department
Queen's University
Kingston, Ontario, Canada
Email:ibnkahla@queensu.ca

***Abstract*—With the rise of internet of things an immense number of RFID tags will be associated with different systems that require not only strong authentication protocols, but also time- and power- efficient protocols to authenticate more tags in a given time window. In current tag authentication protocols, a tag is considered authentic if the interrogators find a match to the tag's encrypted (e.g., using some hashing function) reply in the system's database. Tree-based authentication protocols provide rapid authentication by limiting the searched keys at the interrogator from $O(N)$ to $O(log(N))$, where N is the number of leaves in the balanced tree. However, if one tag is compromised in such protocols, other tags will be at risk of being compromised. In this paper we propose Unique Hash Sequence Authentication (UHSA) protocol. The protocol utilizes tag-interrogator interaction, with a continuous wave (CW) sensor at the tag to cut off tags encrypted reply when the received bits are enough to determine next node in the tree without receiving the whole reply. Cutting off the encrypted reply limits the information that can be obtained by the adversary to compromise the tag. In addition, the reduction in tag reply length greatly enhances the time and power efficiency of the RFID system during the authentication process by more than 90% when compared to existing authentication protocols.**

I. Introduction

RFID technology is an attractive solution for massive tagging of objects over other identification technologies due to the long reading range and data storage capabilities. With these attractive capabilities, guaranteeing privacy protection is a crucial step in the public adoption of tagged objects towards internet of things realization. In addition, tags that are immune to compromising and tracking attacks are essential in several applications (e.g., retail, pharmaceutical and medical application).

In RFID systems, an adversary might obtain the tags ID or part of it's memory content in the absence of privacy protection protocol. Therefore, several Privacy Preserving Authentication (PPA) procedures have been proposed [1]–[4]. In PPA, the interrogator sends a challenge message to the tag, and the tag replies back with an encrypted message that contains an authentication key. The interrogator then searches its database for the key that matches the one sent by the tag. Unauthentic interrogator is unable to decrypt the message and obtain the key as it has no access to the keys database. However, most of the existing PPA protocols many encrypted keys is being transmitted from the tag to the interrogator during each authentication; which consume the inerrogator power and extend the authentication time if many tags are to be authenticated. In addition, PPA protocols that utilize smaller number of keys will suffer from tracking and compromizing attacks.

In this paper, we propose Unique Hash Sequence Authentication (UHSA), a tree-based PPA protocol. The proposed protocol prevents compromising and tracking attacks and significantly reduces the authentication message length from tag to interrogator. The interrogator turns its Continuous Wave (CW) signal OFF for a predefined period once it receives enough bits from the tag that enable it to distinguish, uniquely, the next node in the tree that it should traverse. The tag, on the other hand, has a Continuous Wave (CW) sensing circuitry. If the CW is "ON", the tag keeps backscattering the encrypted message; otherwise, it discontinues the backscattering of any further data. Since the tag only reveals a portion of the encrypted message, the possibility of decrypting the tag's message, by an adversary, is eliminated. Additionally, the time and power needed for authentication is considerably reduced with the need for only short bit sequences (2% to 5% of the total message) when traversing one node to another in the keys tree.

The remainder of this paper is organized as follows. In Section II, the related work is discussed. The system model is described in Section III, and we propose our protocol in Section IV. The performance is evaluated in Section V followed by the conclusion and the future improvements in Section VI.

II. Related work

Recently, several RFID authentication protocols have been proposed [1]–[4]. HashLock [4] is the primary method for authenticating the tags message at the interrogator. In HashLock, the RFID interrogator generates a nonce sequence (r_1) and sends it to the tag. Consequently, the tag hashes r_1 with its unique key, and the resultant message digest is sent back to the interrogator. The interrogator hashes all the keys with r_1 and matches those hashed values with the received digest from the tag. If a match is found, the tag is considered

978-1-4673-2129-7/12/$31.00 ©2012 IEEE

authentic. The message length in HashLock is short; however, the interrogator searches the whole database of N keys, with time complexity of $O(N)$, to find the match. This is not feasible in systems that have thousands or even millions of tags in its database. Therefore, several protocols have been proposed towards increasing key search efficiency. Tree-based protocols [1], [3] organizes the keys in a tree structure to reduce the searching complexity from $O(N)$ to $O(logN)$. The tree is structured such that each node hold a key (or index of its location) and each tag stores a path keys (which may be shared with more than one tag) and a unique key (the leaf). For authentication, the tag sends the encrypted path keys and leaf key to the interrogator. Then the interrogator performs depth first search to match the received keys with the ones in the database.

Since more than one tag share internal path keys, if a tag is compromised by an adversary, several other tags become vulnerable. In a binary balanced-tree with 2^{20} tags, L. Lu *et al* [3] show that any tag can be compromised with a probability of about 90% if only $log(N) = 20$ tags were compromised. To overcome the compromising attack, Strong and lightweight RFID Private Authentication protocol (SPA) [2] implement dynamic updating of the keys (internal and leaves). In SPA, the interrogator updates the keys of the tag and assigns different path and leaf keys after each authentication process. SPA reduces the probability of locating a tag via compromising attacks to 60% if 20 tags is compromised [3]. To overcome tag compromising attacks, Li Lu *et al.* [3] proposed Anti-Compromising authenticaTION protocol (ACTION). Although ACTION protocol is limiting compromising attacks and reducing the keys stored in the tag to two keys (path key and leaf key), it fails to offer protection against tracking attacks. The internal nodes in ACTION are indices from 0 to δ, where δ is the branching factor from each node in the tree. Based on the path key received from the tag, the adversary can simply hash all the δ values of those indices and traverse the tree.

Our protocol exploit a modified Modulation Silencing Mechanism (MSM) [5] circuitry at the tag, in addition to a matching procedure at the interrogator. The proposed protocol not only prevents both compromising and tracking attacks by limiting available data to the adversary, it also reduces the number of bits sent by tag to the interrogator, hence achieving faster tag authentication and saving interrogator RF power.

III. System model

We adopt tree-based authentication for its efficient searching time and scalability. The interrogator is assumed to have a direct access to the keys database while communicating with passive (or semi-passive) tags through backscattering. The clock speed of at interrogator is assumed to be three orders of magnitude higher than the one of the tag [6], [7] (e.g., 1-2 MHz at tag and 1GHz at interrogator). The same for data rate, the tag-to-interrogator rate is in the order of a hundred Kbps, while it is in the order of few Mbps between the interrogator and database. The key database is constructed of a tree with a branching factor δ as shown in Fig.1, where each node has δ children nodes in the next level. The root (level 0 in the tree) is a common key that is shared with all tags, and the internal nodes from the root to the leaves are internal keys that determine the path for the tags keys. The depth of the tree, d, determines the number of stored keys in the tag's memory (the path keys from root to leaf keys) that allow the interrogator to traverse into the depth of the tree to reach the leaf key.

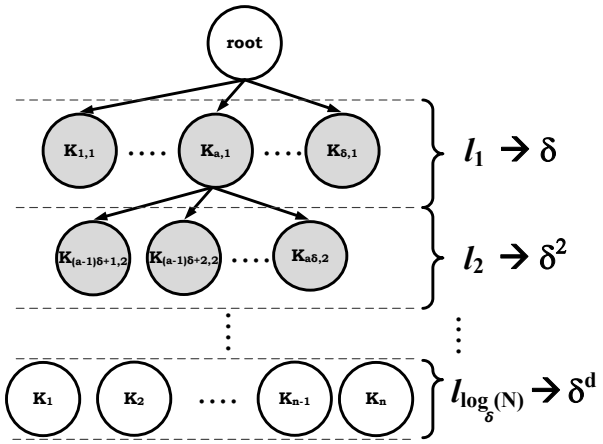

Fig. 1. Key organization in the database

In our design, the interrogator will start authenticating a given tag by sending a random number r_1 and receiving another random number r_2 (generated by the tag). Then the tag hashes d keys from the root key to the leaf key. SHA-1 cryptographic hash function is assumed at the tags and the interrogators with 160-bits output. The d hashed values are represented as $h(r_2, r_1, K_{l,i})$ where l is the level in the tree and i is the index of the key within the δ children nodes. The tag sends $h(r_2, r_1, K_0)$,$h(r_2, r_1, K_1)$,,$h(r_2, r_1, K_{d-1})$, where K_1 is one of the δ children nodes of K_0, K_2 is one of the δ children nodes of K_1, etc.

Once the d values are received, the interrogator starts matching them to the keys in the database when encrypted with the same r_1 and r_2. The interrogator hash K_0 and match it with $h(r_2, r_1, K_0)$ from the tag, if a match is found, it hashes all δ children of K_0 with r_1 and r_2 and matches them to $h(r_2, r_1, K_0)$ from the tag, etc. If the leaf key, the final key, matches on of the hashed values at level $d - 1$, the tag is considered authenticated. Consequently, if further transaction with that tag is required, the leaf key might be used as a secret key in full-fledged cryptography between the tag and the interrogator.

IV. Unique Hash Sequence Authentication

The proposed protocol is based on the cooperation between interrogators and tags to insure synchronization and successfully authenticate each other. Our thesis is if the interrogator is able to determine the next internal node in the key path (out of δ paths for each hashed key) based on receiving part of the hashed value of that node from the tag, there will be no benefit from receiving the rest of the hash value.

In the following, we present an overview of Modulation Silencing Mechanism, followed by the detailed Unique Hash Sequence Authentication (UHSA) algorithms at the interrogator and the tag.

A. MSM overview

Modulation Silencing Mechanism (MSM) allows the interrogator to inform the tags of the occurrence of a collision by cutting off its CW transmission. The tags detect this cut-off and stop modulating their data. MSM components are depicted in Fig.2. MSM-enabled tags sense the interrogators signal availability by the Continuous Wave Absence Detection (CWAD) circuitry. The interrogators, on the other hand, check the received bit streams from the tags for any decoding violations (i.e., not logic 0 or 1). The CWAD circuitry will interrupt the backscattering process by asserting the Backscattering Termination and NACK (BTN) signal.

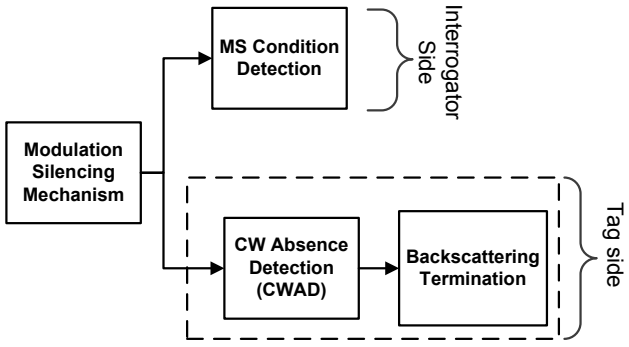

Fig. 2. Modulation Silencing Mechanism (MSM) components

Modulation silencing is activated by the interrogator based on a predetermined condition(s) to end tags reply. The ending condition of the tag reply is not limited to detecting violations in the decoded bit, however, as will be explained shortly; in our authentication algorithm we exploit the ability of stopping the tag reply by the condition of finding a match to the encrypted message from the tag.

Algorithm 1 MSM Algorithm

1: **During Tag(s) reply**
2: **while** *condition= False* **do** ▷ the condition can be any programmed interrupt at the interrogator
3: CW is "ON", Decode received bits
4: **end while**
5: **Stop** CW for a predetrmined period
6: **Send** proper command

At the tag side, MSM detect CW shutdown by the interrogator. Dual antenna tags [8]–[11] will employ MSM without compromising the tag's reading range. Since the interrogator-to-tag link is a Forward Link Limited (FLL) [12], the reading range is dictated by the power received at the tag rather than the power received at the interrogator.

B. UHSA at the interrogator

In UHSA, the interrogator executes the following tasks to insure the synchronization with the tags: Initialization, Pre-fetching keys from the database, Hashing the δ (children) nodes of current node, Determining the next node among the δ nodes. As shown in Algorithm 2, the interrogator starts the authentication process by sending an authentication command to a specific tag based on its random number (R). The random number address is obtained through an earlier singulation (anti-collision) process (e.g., in EPC standard [13], this random address is called RN16). Subsequently, the interrogator initialize the authentication process and send a random nonce (r_1) to the tag.

The interrogator pre-fetches the keys of the next q levels in the tree. The value of q is limited by the storage capacity of the interrogators buffer (or memory) where the total number of pre-fetched keys can be given as: $\sum_{i=1}^{q} \delta^i$. In Fig.3 an example of interrogator at level 0 in a binary tree (δ=2) and q=3, it will fetch δ keys of level 1, δ^2 nodes from level 2, and δ^3 keys from level 3. In general, pre-fetching can be considered a sliding triangle in the tree.

In pre-fetching, the interrogator hashes the next δ keys while it recieves the response from the tag on a relatively slow data rate in tag-interrogator link. During the pre-fetching process, the tag sends the random number r_2. Once r_2 is received, the interrogator stops CW for T_2 commanding the tag to start sending its d hashed keys ($d-1$ path keys and one leaf key).

At the i^{th} hash ($h(r_1, r_2, K_i)$) from the tag, the interrogator hashes δ keys of the i^{th} level in the keys tree. Then the interrogator perform bit-wise matching of the received bits to the δ pre-calculated[1] hashes as shown in 4. The bit-wise matching is performed until only one sequence is matching the received bits (i.e., variable S in 2 is 1). If the received bits do not match any hashed key, the authentication process is interrupted. Higher branching factor, δ, will increase the expected length of the unique sequence (u) as $u = log_2\delta$. This is if the hash values are assumed to follow a random pattern. For example, if δ is 16, the expected values of the first bit in the 16 pre-calculated hash values is 8-zeros and 8-ones. Hence, if the first bit from the tag is one, half of the 16 values will be excluded (all the 8 hashes that start with zero).

The sequence of received bits that matches a single hashed value is called a unique hashed sequence (UHS). After detecting UHS, there will be no need to receive the rest of the hashed key from the tag. This reduces the number of bits that are sent by the tag from 160 bits to $log_2\delta$ bits. However, to prevent an adversary from attaching the tag by not silencing it to receive the whole hash value, the tags are designed not to send more than 50% of its hash values even if CW is still "ON".

Next, if i is less than d, the reader stops its CW for

[1]Recall that the data rate of the tag is around 100Kbps, with a bit duration between 25-8μs which is enough time to compare δ bits at a running clock of hundreds of mega hertz (bit duration of 1ns for 1GHz interrogator processor).

T_2, which commands the tag to send next hashed key ($h(r_1, r_2, K_{i+1})$) and pre-fetch δ^q keys from keys tree that follow the children of the UHS key as shown in Fig.3 . Otherwise, if i =d, both tag and interrogator assume a successful authentication process.

Based on our assumption of three orders of magnitude difference in running speed of the clock between the interrogator and the tag, the hashing time of δ keys by the interrogator is much faster than hashing a single key by the tag (typical δ values are between 2 (binary trees) up to 100 [1], [3]).

Algorithm 2 UHSA at the interrogator

1: **Send** Authentication command to the tag
2: **Send** r_1, **Initialize** level in the key tree (i=0)
3: **Pre-fetch** keys from level l_i to l_{i+q}
4: **if** r_2 is recieved **then**
5: **Stop** CW for T_2
6: **Encrypt** keys of level l_i with r_1 and r_2
7: $j = 1$, $S = \delta$
8: **while** $S > 1$ **do**
9: **if** $j = M/2$ **then**
10: **Stop** CW for T_1
11: **go to** 1
12: **else**
13: **Decode** $h(r_1, r_2, k_i)[j]$
14: **Match** $h(r_1, r_2, k_i)[j]$ to the j^{th} bit of S encoded keys
15: $S = S - \|unmatched\ hashes\|$, $j = j + 1$
16: **end if**
17: **end while**
18: **if** $i = d$ **then**
19: Success, **go to** 1
20: **else**
21: **Stop** CW for T_2, $i = i + 1$
22: **Add** l_{i+q} to Pre-fetched keys
23: **go to** 6
24: **end if**
25: **end if**

C. UHSA at the Tag

In our protocol the tag circuitry is modified to perfrom the actions required by the interrogator algorithm. After receiving the interrogators authentication command and r_1 and start emitting the CW, the tag will generate the nonce r_2 and calculate the hash values of the d path keys. The dual antenna tag [9], [10] uses one of its antennas as shown in Fig.5 to send its data by backscattering the interrogators CW. The second antenna will be used to sense the availability of the CW by introducing a UHSA circuitry (shown in Fig.6) after the second rectifier. If the tag is not addressed by the interrogators authentication commands, both antennas will be used to harvest the power of the interrogators CW.

The purpose of the modification on the dual-antenna design is to allow the tag to detect the interrogators CW shutdown. UHSA at the tag is illustrated in Fig.6. The circuit comprises

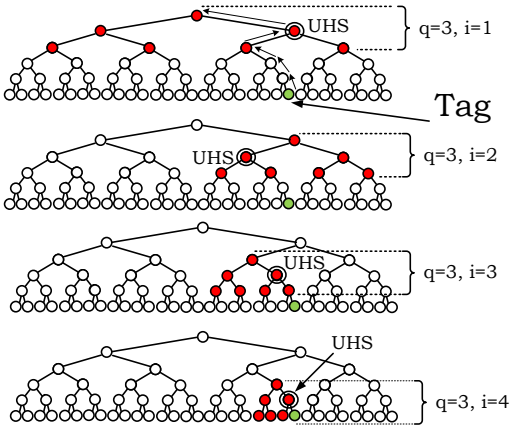

Fig. 3. Pre-fetching example in a binary tree

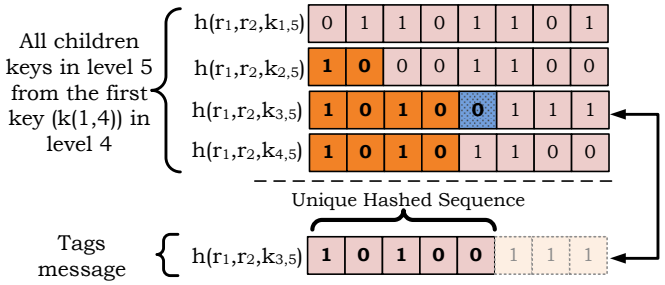

Fig. 4. An example of δ=4 and one tag is sending its 5^{th} hash value. The 1^{st} bit from the tag excludes one hash from the pre-calculated δ hashes, the 2^{nd} is common by the remaining 3 hashes, the 3^{rd} excludes one hash, the 4^{th} is common by the remaining two hashes and the 5^{th} determine the unique sequence.

a voltage limiter after the rectifier to insure a constant voltage to the circuitry regardless of the distance of the tag from the interrogator. The circuit consist of two envelop detector cells (RC circuits), namely C_1 and R_1 circuit and C_2 and R_2 circuit. The circuits is separated by the diodes D_1 and D_2 to allow independent discharging times.

When the tag receive the authentication command, the two

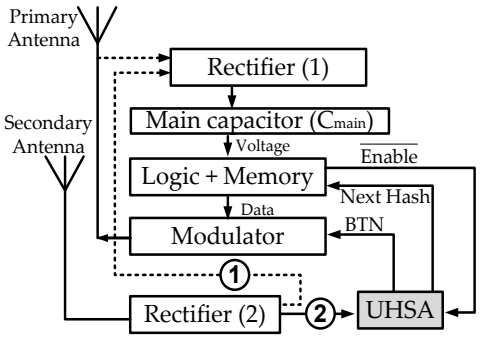

Fig. 5. Dual antenna tag with UHSA circuit, 1- Forward link (tag harvest power by both antennas) 2-backscattering by primary antenna and sensing the CW by UHSA circuit that is connected to the secondary antenna

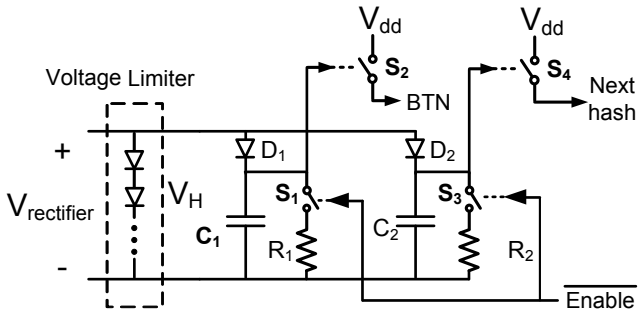

Fig. 6. A schematic of basic circuit components of UHSA at the tag

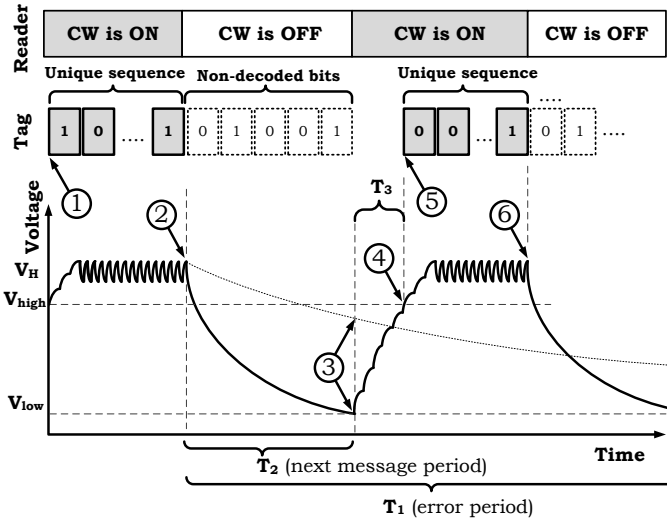

Fig. 7. An example of sending two hash values from the tag in addition to the voltage across C_2 over time.

circuits are activated through two active low switches S_1 and S_2 by $\overline{Enable}$ signal and will follow the algorithm depicted in Algorithm 3. The first circuit, C_1 and R_1, detects long cutoff of CW when the interrogator experience an error during authentication (error in decoding some bits, collision between tow tags) to terminate backscattering permanently until further command. The second circuit, C_2 and R_2, detects short cutoffs of interrogators CW which indicates sending the next hash by the tag. In the design $R_2 \ll R_1$ and $C_2 = C_1$. The falling time from V_H to V_{low} across C_1 is the error period T_1, where V_{low} is the voltage at which S_2 is turned "ON". The falling time from V_H to V_{low} across C_2, denoted as T_2, is the CW cutoff period from the interrogator to indicate sending next hash.T_1 is designed to be much larger than the next hash period T_2. In our simulations, $T_1 = 10T_2 = 50T_{ari}$, where T_{ari} is the bit duration in the reverse link from tag to interrogator in EPC standard [13]. After T_2 the interrogator resume its CW transmission and the voltage across C_2 will start building up in a period denoted as T_3. Once the voltage across C_2 is V_{high} the tag will start sending the next hash.

An example of sending two hash values from the tag is given in Fig.7:1- Tag start sending a hashed value to interrogator. 2-

Algorithm 3 USHA algorithm at the tag

1: **Decode** r_1, **Initialize** key sequence index (i=0), **Assert** $\overline{Enable}$ (S_1, S_3 are ON)
2: **Send** r_2
3: **while** ($S_4 = OFF$ & $j \leq M/2$) **do**
4: **Send** $r_1, r_2, k_i[j]$
5: $j = j + 1$
6: **end while**
7: **if** $i = d$ **then**
8: **Wait** for success ACK
9: **else**
10: $i = i + 1$
11: **loop** ▷ infinite loop
12: **if** (S_4 & S_2 are ON) **then**
13: **Reset**
14: **else if** (S_4 & S_2 are OFF) **then**
15: **go to** 3
16: **else**
17: **Wait**
18: **end if**
19: **end loop**
20: **end if**

Unique sequence is detected and CW is turned off and voltage start dropping at C_2. 3- The voltage across C_2 is low enough to turn S_4 "ON" (Note that the high voltage across C_1 is keeping S_3 "OFF"). 4-The voltage is high enough to turn S_4 "OFF" and the tag will start sending next hash. 5 and 6 are similar to 1 and 2, respectively.

V. Performance Evaluation

The proposed protocol targets the robustness of the authentication procedure against compromising and tracking attacks in addition to the time and power efficiencies. The tag is instructed to terminate its transmission once the interrogator determined the key based on the given unique sequence. In fact, to ensure a sufficient level of robustness, if the interrogator did not silence the tag after sending 50% of the hash value, the tag resets its state and assume an error. Therefore, an eavesdroper to a UHSA authentication session between the tag and the interrogator does not obtain enough information even if an exhaustive search is performed. Most importantly, even if a group of tags are compromised and part of the path keys are revealed, the adversary will not be able to compromise the rest of the tags.

In terms of time efficiency, the proposed protocol provides a superior time and power savings by limiting the expected length of the tag message (for each key) to $log_2\delta$. Different values of δ and tree depth d are plotted in Fig.8. The hash function is assumed to be SHA-1 with 160 message length ($|h|$) for each key. The random numbers r_1 and r_2 are assumed to be 80-bits as in [1], [3].

Performance comparison of state of the art PPA authentication protocols is given Table I. UHSA message length is $r_2 + d * log_2(\delta)$ which is much shorter than any other PPA

protocol. The computation complexity at the tag is similar to [1] where each tag calculate d hashes in each authentication session. In [3] the computation complexity at the tag is better than UHSA, however, it is vulnerable to compromising attack where an adversary can easily extract path key by enumerating r_1 and r_2 with $[1, 2...\delta]$ to define the possible indices of next nodes. The computation complexity of UHSA at the interrogator is defined by hashing the children keys of the previous key for d times as in [2]. In addition, a bit compare process is performed d times. In [3] the interrogator perform only δ hashes and that make it vulnerable to extraction attack by eavesdropping to only one tag-interrogator transaction.

Even for high δ values, $log_2\delta$ is much lower than 160-bits. Therefore, a significant time and power is expected in the overall authentication session. In fact, UHSA message length is shorter than non-tree based protocols while achieving the scalability of tee-based protocols. In Fig.8 a plot of the total message from the tag to interrogator is depicted for different δ and tree depth values. For instance, a tag in a tree depth of 8, and δ of 20 (the tree contain $1.28 * 10^9$ leaves) is expected to be authenticated with an 120 bits. On the other hand, for the same tree SPA [2] and ACTION [3] requires a total message of 1360 and 400 bits respectively; a saving of more than 91% and 70/

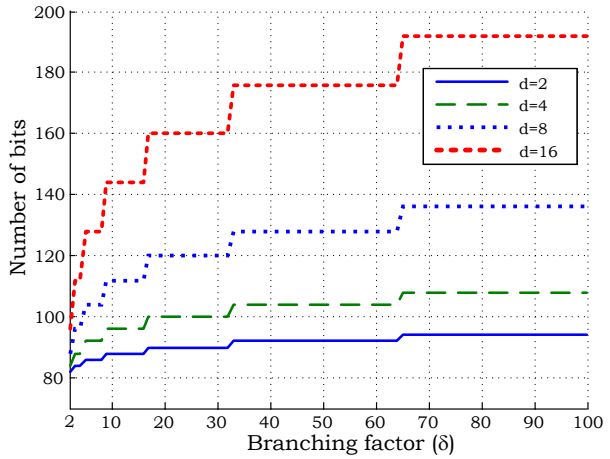

Fig. 8. Number of bits transmissted by a single tag during authentication process.

UHSA achieves a significant reduction in message length while maintaining a robust authentication process for compromising attack by not revealing the whole hash values and to tracking attack by sending different reply for every authentication message. In addition, the proposed protocol is scalable and the modification on the tag design is minor when compared to the complex design of the logic, memory and rectifier circuitry of the tag.

VI. Conclusion

In this paper, we increased the robustness and efficiency of tree-based Privacy Preserving authentication through Unique Hash Sequence Authentication (UHSA) protocol. The interrogator in UHSA stops tag's reply once it receives sufficient number of bits (Unique sequence) that enable the interrogator to select the next node in the balanced-tree database. In UHSA, the interrogator stops the tag reply by utilizing a modified Modulation Silencing Mechanism (MSM) which emulates a full duplex link between the interrogator and the tag. UHSA reduces the time of authentication messages while maintaining the privacy protection from compromising and tracking attacks. The significant reduction of tag's overall message length increases the time efficiency of the authentication process and limits the amount of data available to the adversary. Compared to previous PPA protocols, UHSA prevents both compromising and tracking attacks with more than 90% reduction in the overall tag-interrogator message length.

TABLE I
Performance comparison between UHSA and state of the art PPA protocols

	Tag message length	Compromising /tracking attacks	Scalablity	No. of keys in the Tag
Sarma [4]	$r_2 + \|h\|$	Robust/ Weak	No $O(N)$	1
SPA [2]	$r_2 + d * \|h\|$	Weak/ Weak	Yes $O(d * \delta)$	d
ACTION [3]	$r_2 + 2 * \|h\|$	Robust/ Weak	Yes $O(\delta + 1)$	2
UHSA	$r_2 + d * log_2\delta$	Robust/ Robust	Yes $O(d * \delta)$	d

Acknowledgment

This research is funded by a grant from Natural Sciences and Engineering Research Council of Canada (NSERC)

References

[1] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, Mar. 2006, pp. 6 pp. –275.

[2] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic key-updating: Privacy-preserving authentication for rfid systems," in *Pervasive Computing and Communications, 2007. PerCom '07. Fifth Annual IEEE International Conference on*, Mar. 2007, pp. 13 –22.

[3] L. Lu, J. Han, R. Xiao, and Y. Liu, "ACTION: Breaking the privacy barrier for RFID systems," in *INFOCOM 2009, IEEE*, Apr. 2009, pp. 1953 –1961.

[4] S. E. Sarma, S. A. Weis, and D. W. Engels, "Rfid systems and security and privacy implications," in *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES '02. London, UK, UK: Springer-Verlag, 2003, pp. 454–469. [Online]. Available: http://dl.acm.org/citation.cfm?id=648255.752715

[5] A. Y. Alma'aitah, H. S. Hassanein, and M. Ibnkahla, "Modulation silencing: Novel rfid anti-collision resolution for passive tags," in *RFID (RFID), 2012 IEEE International Conference on*, Apr. 2012, pp. 81 –88.

[6] D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice*. Newton, MA, USA: Newnes, 2007.

[7] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards and identification*, 3rd ed. John Wiley & Sons Ltd, 2006.

[8] D. Deavours, "A circularly polarized planar antenna modified for passive UHF RFID," in *RFID, 2009 IEEE International Conference on*, Apr. 2009, pp. 265 –269.

[9] H.-J. Li, S.-Y. Chen, and D.-S. Yuan, "A novel dual-antenna structure for UHF RFID tags," in *Antennas and Propagation Society International Symposium, 2009. APSURSI '09. IEEE*, Jun. 2009, pp. 1 –4.

[10] P. Nikitin and K. Rao, "Performance of RFID tags with multiple RF ports," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, Jun. 2007, pp. 5459 –5462.
[11] Impinj, *Monza/ID Preliminary Data Sheet, rev. 1.3*, May 2007.
[12] R. Chakraborty, S. Roy, and V. Jandhyala, "Revisiting RFID link budgets for technology scaling: Range maximization of RFID tags," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 59, no. 2, pp. 496 –503, Feb. 2011.
[13] *EPC Radio-Frequency Identification Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860MHz-960MHz, EPCglobal,*, Std., Rev. 1.2.0, Oct. 2008.