A Two-Step Mutual Authentication Protocol Based on Randomized Hash-Lock for Small RFID Networks

Kaleb Lee [1]
Faculty of Science and Technology, QUT
kaleb.lee@student.qut.edu.au

***Abstract*—RFID has been widely used in today's commercial and supply chain industry, due to the significant advantages it offers and the relatively low production cost. However, this ubiquitous technology has inherent problems in security and privacy. This calls for the development of simple, efficient and cost effective mechanisms against a variety of security threats. This paper proposes a two-step authentication protocol based on the randomized hash-lock scheme proposed by S. Weis in 2003. By introducing additional measures during the authentication process, this new protocol proves to enhance the security of RFID significantly, and protects the passive tags from almost all major attacks, including tag cloning, replay, full-disclosure, tracking, and eavesdropping. Furthermore, no significant changes to the tags is required to implement this protocol, and the low complexity level of the randomized hash-lock algorithm is retained.**

I. Introduction

Radio Frequency Identification (RFID), as the name suggests, is a method of identification utilizing the transmission and reception of electromagnetic or electrostatic radio waves [1]. The technology stems from the principle of remote storage and retrieval of data, using devices called transponders and readers. In its simplest form, the transponder, commonly in the form of a tag, emits radio signals when it is within the range of the electromagnetic field generated by the reader [2]. The reader then picks up the relevant signal, which contains information about the object attached to the tag [3] [4].

The potential advantages, including cost savings, brought possible by the use of RFID would likely make RFID tags one of the most commonly deployed microchips in history [5]. Like any widely deployed technology however, RFID has also attracted the attention of attackers aiming to exploit RFID for non-legitimate use. As the powerful technology can identify objects without the line-of-sight requirement, there is an urgent need for developing effective security measures to protect RFID tags from compromising confidential information.

The randomized hash-lock protocol, unlike other hash-based protocols such as hash-lock and hash-chain [6] [7], has received little or no enhancement since it was proposed. Perhaps this phenomenon is due to its practicability (compared to other hash-based scheme). Unlike the scheme in [8] and the new protocol proposed here, almost all other authentication schemes does not allow for mutual authentication to be completed in just two message exchanges, but at least three to four exchanges [9] [10] [2] [11] [12] [13] [14] for one-sided authentication alone. Aside from the above advantage, the randomized hash-lock protocol also exhibit better performance over other protocols when used in small RFID networks where there is only a small amount of tags. The new two-step authentication protocol proposed thus also best suit situations where only a relatively small amount of tags are involved.

This paper proposes an authentication scheme that addresses anonymity, authentication, and confidentiality to some extent, based on the randomized hash-lock scheme proposed by Wies et al [8]. We aim to address several most significant vulnerabilities of RFID technology, such as replay attacks and full-disclosure. It should be emphasized that authentication is especially important to the tags employing these protocols, as it is their functionality that is of most interest to the attackers.

II. Randomized Hash-Lock

One of the first proposed, and most frequently discussed and enhanced protocols ([6] [7] [15].) is the Hash-Lock protocol proposed by MIT [8], it has been used as a foundation for many other protocols such as the Enhanced Authentication Protocol Based on Hash-Lock proposed by Ouyang et al [15]. Its low complexity and efficiency has attracted many researchers to use hash-lock as a starting point of their research.

The randomized hash-lock was proposed along with the hash-lock protocol by Weis [8] in 2003. The introduction of a random number R introduces an additional layer of complexity to the hash-lock scheme. In order to randomize output of the tag for every session, the response of the massage has been lengthen from $H(id_i)$ to $H(id_A, \|R)$, where H, R and id are the hash function, random number and id of the tag respectively. As R is a different a random number generated for every session, $H(id_i, \|R)$ would not be as predictable compared to $H(id_i)$. Similar to $H(id*)$ used in hash-lock, $H(id_i, \|R)$ is used as a challenge to the reader's queries. The reader has to return the tag's ID as a proof of its legitimacy.

An authentication session is started at the query for $H(id_i, \|R)$ from the reader, to which a tag has to first generate a random number R from its embedded random number

[1] MCSE:Security, MCSE, MCSA:Security, MCSA, RFID+, Security+, MCITP,Queensland University of Technology, Australia

generator. Using R, a new challenge message is generated by combining the original ID and R creating $(id_A, \|R)$, which is than hashed creating $H(id_A, \|R)$. $H(id_A, \|R)$ is then forwarded back to the reader as a response. After $H(id_A, \|R)$ has been received by the reader, the reader queries the ID of all tags in the database, each ID is combined with R received from the tag ,creating every possible combination of $(id_i, \|R)$. $H(id_i, \|R)$ is generated for every existing $(id*, \|R)$ until $H(id_i, \|R) = H(id_A, \|R)$ in which case the tag is deemed to be authentic. Consequently if no existing $(id*, \|R)$ matches $H(id_A, \|R)$, the session will end with the reader regarding the tag as illegitimate.

If a tag is deemed to be authentic, the reader will respond to the tag with $(id_i$ where $H(id_i, \|R) = H(id_A, \|R)$ as a response to its initial challenge $H(id_A, \|R)$. If $(id_i$ and $(id_A$ has been compared by the tag and is found to match, the tag would unlock itself allowing the reader to access it full functionality, completing the authentication process.

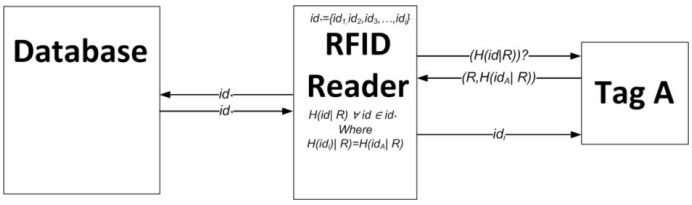

Fig. 1: Randomized Hash-Lock Authentication Process

A. Major Limitations of Randomized Hash-lock

Although randomized hash-lock brings significant benefits into preventing attacks like tracking, its disadvantages may very well outweigh its advantages. One of the most significant changes made that caused deployments to avoid such implementation is due to the amount of resources required for the protocol to operate. A reader has to effectively calculate hashes for records of every single tag ID stored in the back-end database in order to authenticate only one tag. This process has to be repeated every time any tag has to be authenticated, limiting its practicability for all networks except only those with a small amount of tags.

Due to the inclusion of a random number generator, the tag requires larger amount of gates to be implemented into its chip, which in turn increases the price of a tag. But perhaps this increase would not have much effect on the decision of deployment, as the randomized hash-lock scheme would not be a feasible solution in situations that require a large amount of tags allowing the total cost to be relatively low.

Compared to hash-lock where the ID does not necessarily have to be stored in the tag, this scheme requires the ID to be stored in the tag for the authentication process to complete. Being so makes the ID vulnerable to be extracted by attacks such as replay and spoofing attacks. It is now possible to compromise the ID just by eavesdropping a legitimate session, as the ID is sent as a response to the challenge $H(id_A, \|R)$.

The heavy calculations required before a tag can be authenticated makes this protocol especially vulnerable to DoS attacks. By simply authenticating with a reader by randomly generated messages of the required length, the reader would accept each message as an authentication request and consequently consuming a huge amount of resources. With a large enough amount of such messages, a DoS can be effectively launched.

Randomized hash-lock offers little protection against replay and spoofing attacks. Using only two authentication sessions an attacker can extract the ID of a tag, and in turn it's functionlity. In the initial step, the attacker is required to query the targeted tag for $H(id_A, \|R)$, using $H(id_A, \|R)$ an attacker would attempt to establish a connection with a reader. It is important note that, although $H(id_A, \|R)$ is supposed to be different every session there are no measures preventing $H(id_A, \|R)$ to repeat throughout the tag's lifetime. By replaying a captured $H(id_A, \|R)$, the attacker's authentication attempt would still be considered genuine. Taking advantage of this assumption, by replaying a captured $H(id_A, \|R)$ a reader would accept this as a legitimate challenge and therefore respond with the ID of the original tag, which is captured by the attacker for the final step of compromise. The targeted tag's full functionality would be unlocked, when the attacker establishes a new session with the tag, regardless of the tag's challenge respond with its ID and be accepted by the tag. This approach exploits the fact that regardless of the challenge $H(id_A, \|R)$, there can only be one correct response, which is the ID of the tag. By replaying this response an one can access the full functionality of a compromised tag for as long or as much as one wishes.

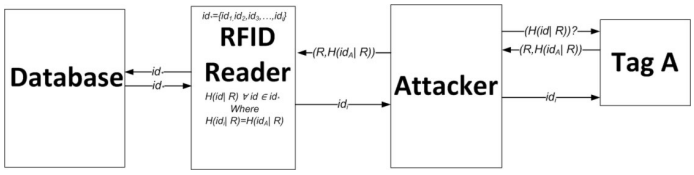

Fig. 2: Spoofing and Replay on Randomized Hash-Lock

III. Proposed Two-Step Mutual Authentication Protocol Based on Randomized Hash-Lock for Small RFID Networks

Similar to the original randomized hash-lock protocol, where mutual authentication can be achieved in two message exchanges [1], this protocol also allows for mutual authentication in two message exchanges but at the same time address possible points of compromise in the original protocol. By using the message $H(R\|id_A)$ instead of id, prevents id to be compromised during the authentication process.

The authentication process of this protocol follows similarly to the randomized hash-lock protocol. Using random number R

[1]Whenever a tag is able to create an authenticable hash and random number combination, it is assumed that the tag is authentic. If the reader is able to respond to its query, replying with its ID, the tag assumes that the reader is also authentic. Although this is by no means a secure process, it does allow for mutual authentication under non-hostile circumstances.

generated using the embedded random number generator at the beginning of every session. Tag A creates message $H(id_A, \|R)$ as a challenge to the reader.

After $H(id_A, \|R)$ has been received by the reader, the reader queries the ID of all tags in the database, each ID is combined with the random number$R$ received from the tag ,creating every possible combination of $(id_i, \|R)$. In order to calculate $H(id_i, \|R)$. $H(id_i, \|R)$ is generated for every existing $(id*, \|R)$ until $H(id_i, \|R) = H(id_A, \|R)$ in which case the tag is deemed to be authentic. Consequently if no existing $(id*, \|R)$ matches $H(id_A, \|R)$, the session will end with the reader regarding the tag as hostile.

If a tag is deemed to be authentic, the reader will respond to the tag with $H(R\|id_A)$as a response to $H(id_A, \|R)$. The reader will be authenticated after the tag has calculated $H(R\|id_A)$ and compared it to the one received from the reader and that they match.

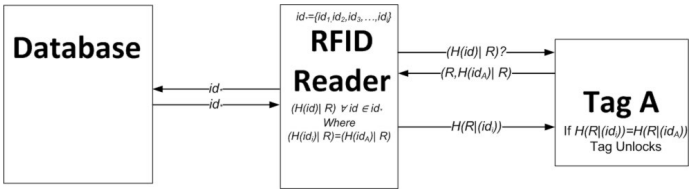

Fig. 3: Enhanced Protocol Based on Randomized Hash-Lock

IV. Protocol Analysis

While many enhancements of protocols, such as the Enhanced Authentication Protocol Based on Hash-Lock [15], implements addition steps or process to the existing protocol, this protocol takes the opposite approach and aims to minimize additional complexity to the original scheme while enhancing security. This is clearly evident in the authentication process of the enhanced protocol. By not implementing any extra authentication steps, or any other mechanisms such as synchronization, this protocol maintains the clarity of its original design by inheriting its basic two-step authentication structure.

Whilst some might believe that authentication protocols with little complexity are inferior to those which have high complexity, it is not always the case. Whereas complex authentication protocols typically offer more features or claim to be more robust, the relatively large amount of steps required for the process to complete creates a larger potential attack surface, not to mention additional cost due to more complex circuitry.

Protocols cannot be considered to be secure if it depends on the obscurity of the choice of hash function, mainly due to the limited choices currently available for low-cost RFID systems. Therefore the protocol is developed under the assumption that all information except the ID of the tag is known by all parties.

The lack of protection for the final message ID sent from the leaves a more secure response that is resistant to replay, spoofing and eavesdropping to be desired. As textitH(R$\|id*$) is generated with the same elements as $H(id_A, \|R)$ using the same process, they are of similar cryptographic complexity. Although it may seem at first that the similarity between $R\|id*$ and $id_A\|R$ seems insignificant, it is not possible to generate $R\|id*$ using only information in the challenge R, $id_A\|R$ even with the knowledge of the hash function H.

As shown in table I, using the SHA-1 algorithm as an example, the results of $H(R\|id)$ and $H(id\|R)$ are vastly different, maybe even impossible do draw any similarities just by looking at the two results. The ID is still used as a 'key' for unlocking a tag, is no longer used as a direct response the challenge, but is rather used to generate a message that requires its knowledge.

Tag ID (id)	da4b9237bacccdf19c0760cab7aec4a8359010b0
Random Number (R)	563
$H(id\|R)$	a2567f91877676585b63b8a04c6aaa0eeb26d4fc
$H(R\|id)$	96db9bd178471821024ca4d1dc3e90f4b37100c4

TABLE I: H(id$\|$R) and H(R$\|id$)

V. Security Analysis

As with any proposed security protocol, it would be interesting to analysis if the proposed protocol offers any real advantage towards existing security threats. Although one would be unrealistic to expect any protocol to be perfectly immune to attacks, especially security protocols for wireless networks, any advantages compared to older protocols should always be welcomed.

A. Eavesdropping

It is not feasible to completely prevent eavesdropping on most wireless networks [16] [17]. Authentication protocols can only limit the amount of information transmitted though the physically insecure communication channel [18]. By limiting as well as encrypting information exchange during authentication, protocols can decrease the value of information gained by eavesdropping and therefore lower the favorability of such attack.

The proposed authentication only requires a total of two message exchanges [2] allowing the authentication process to be completed in a relatively short amount of time depending on the total amount of tags. In addition to the above advantages, both messages exchanged in the authentication process are encrypted using a one-way hash function effectively turning captured messages into useless hashes of randomized strings if captured. Although the proposed protocol cannot be used to control any data leakages after a tag has been unlocked, it is effectively into preventing any information that is of any interest to attackers to be compromised during authentication.

B. Man-in-the-Middle Attacks

Having only two instances of messages exchange between the reader and tag, it is in some way technically impossible to launch a MITM attack anytime during the authentication

[2]Excluding the initial reader query, as it is technically not part of the authentication

process [12]. However as the proposed protocol does not have any control over communication of a tag after it has been unlocked, MITM attacks can take advantage of an unlocked tag after it has been authenticated, as shown in figure 4.

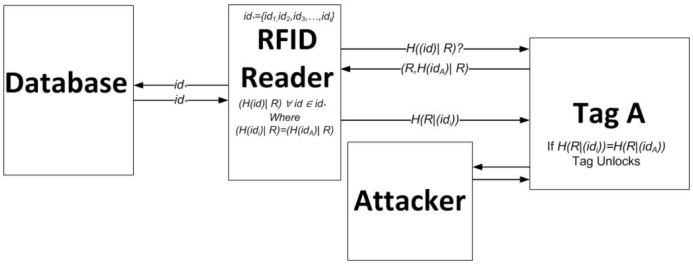

Fig. 4: MITM Attack on the Proposed Protocol

The attack shown in figure 4 is possible mainly due to the absence of a continuous authentication after the initial authentication procedure. A tag assumes that after it has successfully authenticated a reader, the subsequent communications are from that authenticated reader and are therefore authentic.

C. Replay

The proposed protocol can be quite effective into preventing replay attacks. One of the main weaknesses of the randomized protocol is due to the use of a non-changing tag ID as a method of proofing a reader's authenticity to a tag. Its constant nature allows this ID to be captured and replayed for all subsequent authentication sessions, allowing an attacker to continuously compromise a tag for as long as the ID remains unchanged. It is no-longer possible in the proposed protocol as a pseudo-random key $H(R\|id_*)$, derived with the knowledge of both the random number R as well as the ID of the tag, is used in place of the traditional constant ID. Although the proposed protocol has major advantages over the tradition randomized hash-lock protocol in areas of preventing replay attacks on a tag, it does not offer anything mechanisms into enhancing security against replay attacks on readers. It is assumed that the full functionally of a tag is of most interest to an attacker and consequently the possibility of compromising a reader is relatively low.

Interestingly, however, the proposed protocol is not perfect into preventing replay attacks with one drawback. The protocol's security is dependent on the randomness of the random number generator, as one may realize, $H(id\|R)$ and $H(R\|id)$ are in reality key pairs. As the only changing variable of both messages is R. For example, of two sessions, 1 and 2, from the same RFID tag using the proposed protocol, where the random numbers generated are $(R_1$ and $(R_1$ respectively, and consequently two challenge-response key pairs ($H(id_1, \|R)$,$H(R\|id_1)$) and ($H(id_2, \|R)$,$H(R\|id_2)$) would be created. Generally one would expect the values for the two pairs to be different, however one could not ignore the possibility of the same R being generated, effectively reusing a key-pair more than once. As there are no records of previous generated numbers, and the creation of such records could potentially limit number of times tags could be authenticated, the possibility of the same R being generated grows as the number of tags continues to expand.

Although there is a theoretically an infinite amount of combinations possible practical limitations, such as storage requirements and computational requirements, allow only for a finite amount. By capturing a large amount of key-pairs, an attacker can initiate the authentication process until a key generated by a tag matches a previously captured key-pair. However an attacker would have to eavesdrop to capture a considerable amount of key-pairs in order to make this attack effective. Not worthwhile for most attackers.

D. DoS

Not only does this protocol inherit all the advantages of the randomized hash-lock scheme, but also inherits some of the disadvantages as well. DoS attack by flooding is the most notable examples of such disadvantages. Similar to the original randomized hash-lock scheme, the proposed protocol requires the reader to calculate $H(id\|R)$ for every id, as readers cannot distinguish whether a communicating tag exists in the connected back-end database until $H(id\|R)$ has been performed on all existing*id*s.. Although one such hash calculation might not consume much resources on modern systems, a large concurrent amount of such operations would eventually consume all available resources causing the reader the halt operation [19] [20].

By taking advantage of this characteristic, an attacker can easily generate random messages of the required length, with no intention of authenticating with the reader, putting an enormous burden on the reader/database effectively performing a denial-of-service attack.

E. Tracking

Similar to the randomized hash-lock protocol, this protocol can be effectively into preventing identity tracking. By utilizing random response to any reader prior to authentication, it is no longer possible to identify a tag prior to being authenticated. Traditional tracking exploits the non-changing challenge message used for authentication. By introducing randomized authentication challenges, they have become unpredictable to an attacker an effective mechanism into preventing tracking.

However, the above discussed is under the assumption that an attacker does not have any knowledge of the ID of the tag, if an attacker can successfully authenticate with the tag, tracking cannot be prevented. This scenario is common if the 'attacking' party is in fact the owner or provider of the tags, hence they could very possibly have more knowledge about the tag then their current owners.

It is interesting to note that physical tracking is still possible as the tag would still respond to readers with its authentication challenge, allowing the signal to be traced. [11] [21]

F. De-synchronization

The current proposed protocol does not make use of any synchronization values, and hence there are no possibilities

of such attacks under the current scheme [22]. However synchronization values could to be included in the future, as the current protocol have no method of determining whether the communicating reader is authentic after authentication. A synchronization value could possibly be used in order for a tag to keep record of the number of messages since a reader was authenticated.

G. Full-Disclosure

A full-disclosure attack can be regarded as the compromise of the ID stored on the tag, as it is the most vital piece of information required to unlock a tag [14]. As an unlocked tag would assume a communicating reader is authentic, an attacker can possibly extract all information from a tag once it is unlocked.

Although the proposed scheme does not have any obvious points of exploit that allows for a full-disclosure attack though the authentication process, it is possible for an attacker to perform a brute force attack that could reveal the ID of a tag using the initial challenge message *H(id||R)*. By computing all possible values of *(R,H(id||R))* until one finds a match, however In order for this attack to be successful an attacker must first be able to identify the length of the ID as well as the structure of the message *(id||R)*. For example, if the challenge message *(R,H(id||R))* represented by hexadecimal strings *(563, a2567f91877676585b63b8a04c6aaa0eeb26d4fc)*, one would have to know that the original *(id||R)* is in the form of: *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXHHH* where *X*s represents the hexadecimal ID string and *H*s the random number. By hashing every possible combination, i.e from *000000000000000000000000000000000000000000563* to *FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF563*, an attacker would be able to distinguish the ID of the tag.

However it is important to note that multiple strings can produce the same hash. Where the length of the ID is longer than the fixed output length of a hash function it is guaranteed that there would be multiple IDs with the same hash. Hence returning to the fact that the proposed scheme is only suited for deployments where there are only small amount of tags.

H. Cloning

No authentication mechanism can be used to prevent cloning, but some can offer advantages over others into preventing attackers from gathering data required for cloning. In the randomized hash-lock protocol, the ID of the tag is the signal most critical information one needs in order to compromise a tag. As mentioned previously, one of the main concerns over the original randomized hash-lock scheme is caused by the use the tag's ID as a key for unlocking the tag. The ID during transmission is at high risk of being compromised. This risk can be minimized by using a utilizing a hashed message, in place of the original, that requires one to have knowledge of the ID.

However all would fail if an attacker physically possesses the tag. Using a physically invasive attack, an attacker can directly extract all information stored in a given RFID tag if given the correct equipment.

VI. Other Considerations

Aside from security considerations, there also exist practicability concerns. Whilst the proposed protocol offers significant overall advantages over the existing protocol, is still relatively early in its lifecycle, below are two issues that could potentially affect security in the future.

A. Memory Requirements

One of the main drawbacks associated with the currently proposed protocol comes from the fact that it requires nearly twice the amount of memory than the original. In the randomized hash-lock protocol, after the initial challenge message $H(id_A, \|R)$, the tag no longer have the need to store R as the only correct response is *id* and *id* is typically be stored in read-only non-volatile memory, where as generated hash outputs are stored in rewritable volatile memory.

In the original protocol only the *id*, received from the reader, or the initial challenge textit $H(R\|id_*)$ is required to be stored in volatile memory at one time. However, as the proposed protocol uses textit $H(R\|id_*)$ in place of just *id* as a response to the challenge $H(id_A, \|R)$, $H(id_A, \|R)$, R as well as the $H(id_A, \|R)$, calculated by the tag, are all required to be stored into temporary storage memory for the tag to compare them. For example, if the current hash-function produces a fixed length 64-byte output, ID is a 48-byte string and R is a 2-byte random number. The total amount of memory required for the randomized hash-lock scheme is 66-bytes, a 64-byte hash output and a 2-byte random number required for the challenge $(R,H(id_A, \|R))$, which is considered to be the peak consumption for throughout the entire authentication process. Using the proposed protocol, peak consumption is reached when comparing the response $H(R\|id)$ received by the reader and $H(R\|id)$ calculated from the tag, consuming a total of 130-bytes given by a total of 2 hash outputs, one from the reader and one calculated as well as R. The proposed protocol uses consumes a total of 96% more memory compared to the original randomized hash-lock protocol, as shown as a comparison in figure 5.

Fig. 5: Protocol Memory Consumption Comparison

This issue could possibly be avoided by producing shorter hash outputs[3], but doing so would lower the security of the protocol dramatically. It is important to note that, although this is considered to be a major issue now rapidly developing technology will eventually allow larger amounts of memory to be present on RFID tags at lower cost, in which case this would no longer be an issue.

[3]By shrinking the authentication messages by 50%, thus the protocol can be implemented without the need for additional memory.

B. Finite Key Combinations

As suggested earlier on, ($H(id||R)$) and ($H(R||id)$) are in fact key pairs, where only one ($H(R||id)$) exists for every possible value of ($H(id||R)$). There can only be as many combinations as the values possible for R or H depending which has a smaller set of possible values. Although the chances of R repeating in a tag's lifetime are statistically low, one cannot out rule the possibility of deliberate attacks by refreshing R, though authentication attempts, until it repeats.

The chances of R repeating increases after every authentication attempt, as there are one less unique value possible for R. However, this issue can be addressed by periodically changing the ID of a tag throughout its lifetime. By changing ID, a new set of values $H(id||R)$ can be created. For example, by logging every successful authentication in the database a trigger could be created so that id would be refreshed if $H(id||R)$ has been used in previous session, or after every predefined number of successful authentications.

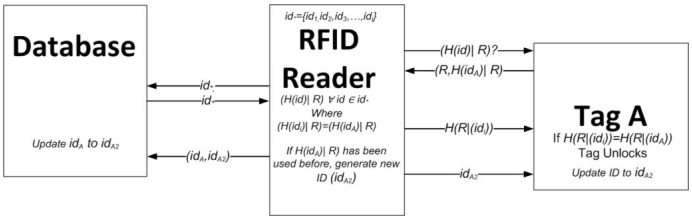

Fig. 6: ID Update Example

Figure 6 outlines the possibility of an ID mechanism discussed in the above example. Observe that the added steps for updating ID adds requirements for additional steps to be implemented in the current scheme. The above is only used as an example, additional work has to be done in order to ensure that the updates do not increase the risk of compromise before such mechanism is implemented.

April 29, 2010

VII. Conclusion and Future Work

The proposed protocol addressees most major security concerns relating to the original randomized hash-lock protocol, especially in preventing replay and full-disclosure attacks. Introducing a unique response for every possible challenge has proven to be an effective measure against replay attacks, which was one of the most vulnerable attacks that randomized hash-lock was subject to. Similar to the original protocol, the proposed protocol retains most of the mechanism for generating random challenges in order to prevent tag tracking.

While the proposed protocol have demonstrated significant improvement in security, however, it has some small limitations. Several aspects such as higher memory consumption and possible, but highly unlikely, coincident reuse of key pairs may slightly hinder its ability to be completely immune to replay and tracking. Nevertheless, aside from these drawbacks, this protocol was able to improve most aspects of security dramatically while remaining free of any unnecessary complexity.

As the proposed scheme is still relatively immature, there are several aspects or ideas that could possibly be developed to enhance the security further. The main ideas of the proposed protocol was to address the non-changing challenge response sent from the reader, allowing the response to be compromised during transmission. Although the protocol no longer uses a non-changing challenge response, it is far from randomized compared to the challenge, as each challenge and response are effectively key pairs. Only by investigating other key generating mechanisms can true random challenge and responses be created.

Finally, future research can look for a method that allow mutual authentication throughout communication, and not just limited to authentication only at the initiation of a session. By requiring authentication after a tag has been unlocked, the risks of a tag being compromised by spoofing after a reader has been successfully authenticated can be significantly reduced.

References

[1] H. Daou, A. Kayssi, and A. Chehab, "Rfid security protocols," pp. 593 –597, dec. 2008.

[2] Z. Luo, T. Chan, and J. Li, "A lightweight mutual authentication protocol for rfid networks," in *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, oct. 2005, pp. 620 –625.

[3] A. Juels, "Rfid security and privacy: a research survey," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 381 – 394, feb. 2006.

[4] Y. Yousuf and V. Potdar, "A survey of rfid authentication protocols," in *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, march 2008, pp. 1346 –1350.

[5] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," *NA*, vol. NA, no. NA, pp. 201–212, 2003.

[6] I. Syamsuddin, T. Dillon, E. Chang, and S. Han, "A survey of rfid authentication protocols based on hash-chain method," in *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, vol. 2, nov. 2008, pp. 559 –564.

[7] T.-L. Lim, T. Li, and T. Gu, "Secure rfid identification and authentication with triggered hash chain variants," in *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, dec. 2008, pp. 583 –590.

[8] S. A. Weis, "Security and privacy in radio-frequency identification devices," 2003.

[9] S. Ahamed, F. Rahman, and E. Hoque, "Erap: Ecc based rfid authentication protocol," in *Future Trends of Distributed Computing Systems, 2008. FTDCS '08. 12th IEEE International Workshop on*, oct. 2008, pp. 219 –225.

[10] T. Y. Won, J. Y. Chun, and D. H. Lee, "Strong authentication protocol for secure rfid tag search without help of central database," in *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, vol. 2, dec. 2008, pp. 153 –158.

[11] N. Park, H. Lee, H. Kim, and D. Won, "A security and privacy enhanced protection scheme for secure 900mhz uhf rfid reader on mobile phone," pp. 1 –5, 0-0 2006.

[12] C. Tan, B. Sheng, and Q. Li, "Secure and serverless rfid authentication and search protocols," vol. 7, no. 4, april 2008, pp. 1400 –1407.

[13] M. B. H. Stephan J. Engberg and C. D. Jensen, "Zero-knowledge device authentication: Privacy and security enhanced rfid preserving business value and consumer convenience," *Second Conference on Privacy, Security and Trust*, vol. NA, no. NA, p. NA, 2004.

[14] T. Li and R. Deng, "Vulnerability analysis of emap-an efficient rfid mutual authentication protocol," in *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, april 2007, pp. 238 –245.

[15] O. Changqing, W. Jixiong, L. Zhengyan, and H. Shengye, "An enhanced security authentication protocol based on hash-lock for low-cost rfid," in *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, aug. 2008, pp. 416 –419.

[16] C.-C. Chen, I.-T. Chen, C.-M. Cheng, M.-Y. Chih, and J.-R. Shih, "A practical experience with rfid security," in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, may 2009, pp. 395 –396.

[17] S. C. Cha, K. J. Huang, and H. M. Chang, "An efficient and flexible way to protect privacy in rfid environment with licenses," in *RFID, 2008 IEEE International Conference on*, april 2008, pp. 35 –42.

[18] Z. Luo, T. Chan, J. S. Li, E. Wong, W. Cheung, V. Ng, and W. Fok, "Experimental analysis of an rfid security protocol," in *e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on*, oct. 2006, pp. 62 –70.

[19] M. Rieback, B. Crispo, and A. Tanenbaum, "The evolution of rfid security," *Pervasive Computing, IEEE*, vol. 5, no. 1, pp. 62 – 69, jan.-march 2006.

[20] A. Sharif and V. Potdar, "A critical analysis of rfid security protocols," in *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, march 2008, pp. 1357 –1362.

[21] H. Lee and J. Kim, "Privacy threats and issues in mobile rfid," in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, april 2006, p. 5 pp.

[22] C. C. Tan, B. Sheng, and Q. Li, "Severless search and authentication protocols for rfid," in *Pervasive Computing and Communications, 2007. PerCom '07. Fifth Annual IEEE International Conference on*, march 2007, pp. 3 –12.