Proxy Framework for Enhanced RFID Security and Privacy

Tassos Dimitriou
Athens Information Technology
Markopoulo Ave., 19002, Peania
Athens, Greece
tdim@ait.edu.gr

***Abstract*—Radio Frequency IDentification (RFID) is a method of remotely storing and retrieving data using small and inexpensive devices called RFID tags.**

In this work we propose a proxy agent framework that uses a personal device for privacy enforcement and increased protection against eavesdropping, impersonation and cloning attacks. Using the proxy a user decides when and where information carried in a tag will be released. In particular, the user can put tags under her control, authenticate requests, release tags, transfer them to new owners, and so on. This is the first framework that unifies previous attempts and presents detailed protocols for all the operations required in such a proxy environment.

Keywords: RFID technology, Privacy, Anonymity, Communications Security, Emerging Wireless Technologies

I. Introduction

Radio Frequency Identification (RFID) is a technology for automated object identification. An RFID tag is an electronic device that is typically attached to an item and upon request transmits item information such as date of manufacture, product characteristics, and so on.

RFID tagged items can have remarkable applications. One could imagine refrigerators issuing warnings about expired food or about remaining bottles of milk. Laundry machines could select washing cycles based on color and sensitivity of clothes. Waiting times at checkout lines could be reduced since RFID readers can scan tags at rates of hundreds per second. Products may be tracked as they move from location to location, improving manufacturing logistics, supply chain management and better inventory procedures. Animals could be retrieved in case they are lost. Even more controversially, RFID chips have been used for “tagging” people, especially school children and club-goers wishing to gain access to VIP areas.

However, the mere fact that communication between tags and readers is wireless and does not require physical contact opens up the possibility for abuse and violation of user privacy. Currently, RFID tags respond to any reader request within range. Consequently, a person carrying a tagged item effectively broadcasts a fixed identifier to nearby readers. Thus anyone with a reader can read the information in the tag, potentially violating the owner’s privacy.

Our contribution: In this work we propose a new framework for enhancing the privacy of users carrying RFID products. Our method uses a mobile phone (or any other similar device) as a *proxy* for interacting with readers on behalf of the user carrying tagged items; thus the user can specify when and where information will be released.

However, apart from enforcing policies and controlling tag state, our method maintains a number of desirable characteristics. In particular, we make sure that only authorized users can acquire and put tags under their control, user access to tags is authenticated, and when a user no longer needs the tagged product she can either make it readable to everybody or transfer it to another user. Furthermore, we make sure that privacy is maintained even if a user’s tag responds directly to scan requests.

II. Design goals

In this section we present a list of general security goals that should be true for any RFID protocol and then proceed with more specific characteristics we expect from our proxy protocol.

Privacy: No secret information should leak from the tag that can help in identifying tag contents or the bearer of the tag. Another closely related problem to that of violating consumer privacy is *location privacy* which may lead to tracking of individuals by the tags they carry. Thus we require that no fixed identifiers should be emitted by the tags or the proxy.

Protection against tag spoofing or cloning: A tag cloning attack would allow a person to either install a replacement tag or simply query the tag and forward its response to a nearby reader. Although, we enable the proxy with the capability to relabel tags to protect user privacy, this should be done in a way that prevents cloning and ensures the privacy of subsequent owners.

Protection against impersonation attacks: An adversary should not be able to impersonate either a tag or the mobile proxy. The first attack could lead to removal of a user’s tagged items since a fake tag could still answer to proxy’s challenges. The second attack could be useful in tracking a user’s movements.

Policy enforcement and access control: We expect the proxy to act as mediator for tag access in order to minimize the privacy risks inherent in the use of RFID technology. So, in certain cases it should be possible to release information about tags while in others to block such requests entirely using proper cryptographic mechanisms.

Transferability and tag release: In many situations it is necessary to bring a tag to its original state (tag release) or transfer it to a new user. In the first case, we require that the current owner should not be able to lie about the original ID of the tag, while in the second we make sure that the privacy of the new owner is guaranteed.

Simplicity and Efficiency: The messages exchanged between tags and proxy should be protected against eavesdropping, impersonation and other attacks. However, this should be done in a way that does not put too much burden on the tags or even inhibit proper tag identification. Finally, we require that the proxy itself be a simple device capable only of communicating with tags and readers, perform various cryptographic operations and perhaps have the available GUI (or other) interface to interact with a user.

III. Related Work

There have been many papers in the literature that attempt to address the security concerns raised by the use of RFID technology. For a survey the reader is referred to [1]. Here we will emphasize on the concept of having users protecting tags they carry by means of personal devices ([2], [3], [4]). The Guardian[2], is a device that acts as an intermediary between tags and readers and *must* always be alert in protecting tag responses from unauthorized read attempts. It has to either allow reader queries, appropriately re-issuing queries in encrypted form, or actively *block* tag answers. Thus if the Guardian fails, security is lost. Furthermore, it does not deal with such issues as tag acquisition and ownership transfer.

The RFID Enhancer Proxy or REP [3] assumes the identities of tags and simulates them in the presence of reading devices by continuously relabelling their IDs. Although similar in spirit to the framework we propose here, the REP suffers from a number of shortcomings such as corruption of tag data, tag-to-REP desynchronization and difficulty in tag release that are attributed to the fact that tag identities need to be partially generated by the tag and match portions of its true ID.

Finally, the authors in [4] propose a scheme called MARP (Mobile Agent for RFID Privacy) that assumes the secrets of tags and uses them to mediate with reader requests *once* the tag has been put to sleep. However, this scheme does not ensure privacy since the back-end server still knows the tag secrets and is always engaged in providing answers to reader requests. Furthermore, the scheme relies on the use of heavy public key cryptography.

The framework we present here unifies the above approaches and offers a complete solution for such issues as tag acquisition, proxy authentication, resistance to privacy attacks, ease of transfer and release, and so on. To the best of our knowledge this is the first work that presents *detailed* protocols for all the operations required in such a proxy environment.

IV. Proxy operations

We now present the protocols that can be used in acquiring and managing a set of tags with the help of a personal device (called *proxy* in the remainder of the paper) such as for example a mobile phone enabled with reader capabilities [5].

In particular, in Section IV-A, we talk about the threat model and the assumptions we make about the capabilities of tags. Section IV-B is the heart of the framework. In this section we explain why it is necessary to relabel tags with new identities, how this can be achieved and what are the issues in doing so. The next two sections are about putting tags to sleep for enhanced protection and waking them up again for possible release. Section IV-E is another important one since it deals with tag responses to *direct* scan attempts. Section IV-F, is about mediation, i.e. how proxy interacts with readers on behalf of the tag, and finally, Section IV-G, is about release and transfer of tags.

A. Threat model and assumptions

Our goal is to protect tag transactions against unauthorized read requests. In general, we assume that the tag shares a *unique* secret K with the back-end database. The unique identifier of the tag does not necessarily serve as a key since this information can be retrieved in numerous ways. So, our threat model does not only include a back-end server which possesses the secret of every available tag.[1] It also extends to users which used to possess certain tagged items that have currently been released to new users. These malicious users through their proxies may engage in all sorts of actions (eavesdropping, replaying messages, interrogating tags, and so on) that may help identify the actions of new users.

In doing so, we are not assuming that tags have strong computational capabilities apart from evaluating simple hash or pseudo-random functions as in many past works ([1]). Additionally, the passive tags in the ISO 14443 family are capable of cryptographic operations.

We also expect the proxy to be capable of interacting with tags and perform several updating operations such as *read* or *write*. A tag may also receive a *sleep* command which will render it inactive unless it is awakened with a *wake-up* command. Unless these operations depend on keyed input, they can be used in tag tracing (more on these operations in Sections IV-C and IV-D). The Atmel TK5552 is an example of a tag that supports the majority of these functions.

Tag ownership: Information about tag secrets can be retrieved at a point of sale. For example, the secret key K can be printed in a receipt in a 2D barcode and have the mobile proxy scan the receipt (see for example [6] and the references therein). Alternatively, the secret can be sent though a wireless channel or a wired one if the proxy can be connected to a transmitting device at the point of sale. However, the wireless method is subject to eavesdropping attacks by other proxies.

[1] This also explains the inadequacy of past solutions that aim to protect users against unauthorized scanning by third parties. What if a coalition of back-end databases is formed (similar to federated identity among enterprises) to exchange information about a user's movements, habits and profile? Thus what is needed is the ability by the users themselves to have complete control of the tags they carry without losing any of the benefits that RFID technology has to offer. This can be achieved by the use of proxy devices.

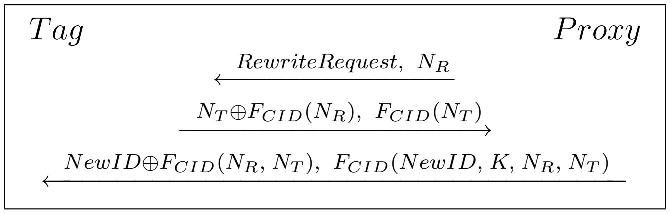
Tag
$Proxy$
$RewriteRequest,\ N_R$
$N_T \oplus F_{CID}(N_R),\ F_{CID}(N_T)$
$NewID \oplus F_{CID}(N_R, N_T),\ F_{CID}(NewID, K, N_R, N_T)$

Fig. 1. Providing a tag with new identity. CID refers to the current ID of the tag, while $NewID$ to the new one.

When this procedure is complete, the user's proxy shares the secret key K with the tag and knows its unique identifier. Notice, however, that this information is available to the back-end server as well. Hence, a network of readers connected to this central location can still use information exchanged between the tag and the proxy to track the user carrying either one of these devices. In the next section, we will describe the heart of our framework, a process where the user can use its proxy to "re-encrypt" the information stored in the tag so that violation of privacy is not longer possible. $F_K(M)$ will denote a *keyed* pseudo-random function applied on a message M and N a random number used once.

B. Tag re-encryption

We refer to the situation where the owner of a tagged product wants to hide the identity of the tag, thus enforcing privacy against tracking and unauthorized read attempts. This is achieved by implanting a *new* identity to the tag which can eliminate unauthorized scanning. Past works [7], [8] have done so by re-encrypting a ciphertext carried by the tag. However, these methods assume that tags have higher computational capabilities. A method, that can be thought as complementary to our approach, is where the tag participates *partially* in the generation of pseudonyms [3]. This method, however, complicates the process of releasing a tag among other things. Instead, our approach relies entirely on the use of the mobile proxy for re-encrypting the tag information which leads to cleaner protocols for tag authentication and tag release.

When the new identity is implanted by the proxy, we no longer need to worry about unauthorized read attempts since the tag will *always* respond using the protocol of Section IV-E, thus making tag responses indistinguishable from random data. The protocol for tag re-encryption is shown in Figure 1. Upon execution of the protocol both the proxy and the tag will update the current ID to $NewID$.

This protocol can be run as many times it is necessary. In all these runs CID refers to the current ID while $NewID$ refers to the newly implanted one. Furthermore, CID is used as a *secret* to authenticate tag and proxy responses. While this is not necessarily true (recall that original identity is also known to the back-end server), it will be true if the *first* tag re-encryption is carried in a private environment (say, one's home). From then on, CID can be treated as a shared secret between the tag and the proxy.

In the message exchange shown in Figure 1, the tag replies with a random number N_T *masked* by the value $F_{CID}(N_R)$, where F is some good pseudorandom function. The last part $F_{CID}(N_T)$ essentially acts as a Message Authentication Code (MAC). Inclusion of N_R in the mask guarantees that both N_T and the response of the tag are fresh. Finally, the new ID is implanted to the tag. The tag first computes $F_{CID}(N_R, N_T)$ and extracts the $NewID$. The second part is used to verify that the new ID was computed successfully by the tag, since if an adversary modifies the first part, the tag would end up with a $NewID'$. However, the last part will not verify and the tag will abort the rewrite operation.[2]

Security analysis: The goal of the protocol is to break the association of both K and CID with a particular tag. If the proxy manages to implant a new identity, privacy is guaranteed even if the back-end server still knows K. This is true because we make sure that the newly implanted ID serves as the new secret for subsequent operations (Sections IV-C to IV-G).

We insist, however, that the *first* rewrite takes place in a secure environment or at least in one where tag-to-reader responses (the *back channel*) are protected from eavesdropping. This is a very weak assumption since as it has been observed by several authors, this channel is much harder to eavesdrop than the forward channel (from reader to tag) [9], [10].

We now proceed to explain why the protocol is resistant to attacks that can be applied by malicious users or proxies.

a) Information leakage and tracking: The scheme guarantees privacy since no fixed identifiers are emitted and no information leaks because of the one-wayness of F. *Semantic* security (or security against chosen plaintext attacks-CPA) is ensured since an adversary (central server or other) cannot compute the new ID from the message $NewID \oplus F_{CID}(N_R, N_T)$ without knowing CID. This is because the generic construct $\langle r, F_K(r) \oplus m \rangle$ is known to be CPA secure. Additionally the protocol is secure against chosen ciphertext attacks since the last parts act as MACs, protecting against modification attempts. The end result, called authenticated encryption, was analyzed in [11].

b) Malicious users issuing re-write commands: Inclusion of the key K in the third step is needed to guarantee that only a *legitimate* owner of a tagged product can issue a rewrite command. If K is omitted, a malicious user can re-install a new ID by first acquiring through eavesdropping the original identity of the tag and then by executing the re-write protocol. As the key K is released only when a user actually buys a tagged product (recall Session IV-A, Tag ownership), a malicious user cannot modify/rewrite the tag without knowledge of K. If a tag receives an unauthenticated read attempt, it will simply abort the modification operation.

c) Malicious owner: What if, however, the real owner wants to clone a tag or swap the identity of a cheap item with that of an expensive one? The motivation for such a behavior would be to enhance the services offered for an

[2] Notice, however, that there is still the possibility of *desynchronization* of the tag from the proxy if an adversary modifies the messages sent by the proxy in the third step. However, we feel that this attack would be difficult to realize since the attacker must know the exact time where step 3 is executed.

already purchased tagged product (an instance of a *swapping attack* [3]). The simplest approach to counter this is to relate the true identity of the tag to the secret key stored in the tag, as suggested in [3]. However, this solution seems very restrictive and in Sections IV-E and IV-G we will see how this can be avoided using only simple cryptographic mechanisms.

d) Spillover: The way the protocol is organized also protects from *spillover* effects. What would happen if *all* tags in the possession of the proxy respond to a rewrite request and eventually change their identities? In our case, this cannot happen since the MAC value in the last step will not verify.

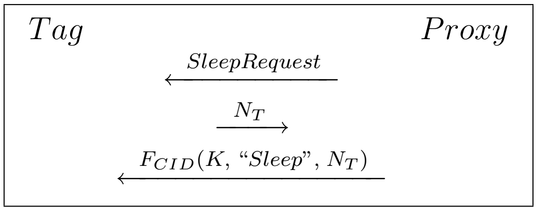

Fig. 2. Putting a tag to sleep.

C. Putting a tag to sleep

Once a new identity is implanted, the owner of the tag can put it to sleep for enhanced privacy protection. Thus the tag will not respond to queries unless of course the tag is awakened again.

The process of putting a tag to sleep can be thought as a simpler alternative to tag re-writing. Again this process must be protected from eavesdropping since otherwise an adversary, by listening to the message exchange between the tag and the proxy, may learn the secret and use it to implant new information or more generally control the tag! If this happens, the proxy will not be able anymore to recognize the tag, thus rendering the tag useless. A protocol that prevents this from happening is shown in Figure 2. Again we rely on the use of the secret ID and a fresh N_T in order to avoid tracing attacks.

D. Awaking a tag

The user, through its proxy device, can use the established secret identity to wake up a particular tag. The need to do so arises when the user has to release the tag to a new user or to the original manufacturer for (say) updated service. The protocol is similar to the one shown in Figure 2 (the only difference is the use of a “Wake-up” command).

E. Masking tag responses

We now focus on protecting tag responses from *direct* scan attempts by any reader. Note that this eliminates the need to enhance the proxy with *jamming* capabilities as in [2].

In the protocol shown in Figure 3, someone (reader or proxy) makes a scan request and the tag always responds with its *true* identity in protected form. *This identity should not to be confused with the current identity CID*, which may be different if planted by the proxy. Since this response is not controllable by the proxy it also serves as a proof that a tag bears its true ID when released to a new user, thus protecting from malicious proxies advertising fake tags and eliminating the need to relate identities to the secret key as in [3].

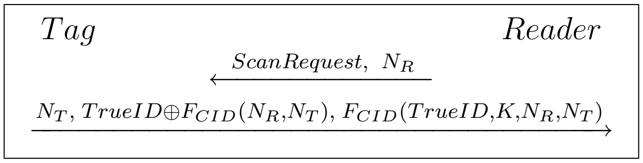
Tag
$Reader$
$ScanRequest,\ N_R$
$N_T,\ TrueID \oplus F_{CID}(N_R,N_T),\ F_{CID}(TrueID,K,N_R,N_T)$

Fig. 3. Enhancing RFID privacy.

The use of the mask $F_{CID}(N_R, N_T)$ helps exactly in blinding this original identity to unauthorized readers. Furthermore, our protocol makes sure that no static code or identity is released by the tag as the masking factor changes with every request (because of the fresh N_T).

By making scan requests, the user's proxy can collect information about the tagged user items since it knows both secrets, CID and K, for each tag. However, the work to be done is *linear* to the number of possessed items since the proxy has to try all possible pairs (CID, K) to see which one is matching the signature $F_{CID}(TrueID, K, N_R, N_T)$ of the second message. This is not really a problem since a user will most likely handle less than a few dozens or at most a few hundreds of tags. For commercial applications, however, we can use scalable versions (like those appearing in [12], [13]), where identification is *logarithmic* in the number of tags.

F. Proxy mediation to tag

So far we have seen that only the proxy that can actually interact with a tagged item and retrieve information about it. In certain cases, however, the proxy must release information about a tag such as when the user must reveal the type of her television set in order to get proper service or even offers about compatible home theaters. Our approach can be thought as complementary to the RFID Guardian [2], where the guardian acts as a “device-in-the-middle” between readers and tags, forwarding cryptographically-protected queries to tags on behalf of untrusted readers. Our proxy, on the other hand, does not forward any requests to the tags; it simply answers on behalf of them. This allows for a wide range of privacy policies: from total release of information to a more refined control about who's the recipient of this information.

In the first case, the proxy simply transmits the original identity of the tag in order to get updated information about services offered, but withholds such information as K and current secret ID. Notice, that doing so does not necessarily prove ownership of the tagged item since the user may just *simulate* tag responses. Hence an adversary cannot be 100% sure that the user has the particular item. In a more extreme case, the proxy can perform selective access control. The proxy may capture requests sent to tags, decode them in real time and determine if the query is permitted.

Finally, the proxy can let the reader speak *directly* to the tag. This is useful, for example, when a washing machine reader must interact with tags carried in clothes in order to select

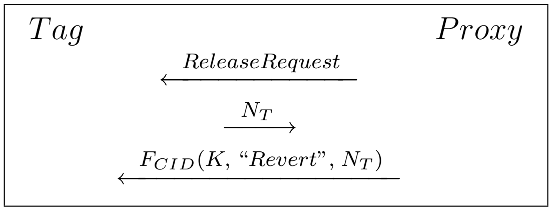

Fig. 4. Bringing a tag back to its original state.

the appropriate program (it would be difficult for the user to actively mediate between the washing machine and the clothes as suggested in other solutions). But how is the reader going to interpret the tags' encrypted answers (recall Section IV-E)? The proxy simple releases both secrets (CID and K) for each tag so that the selected reader decrypts tag responses. Then, at any time, the user can implant a new identity thus refusing further access to the reader.

G. *Tag release*

We conclude the discussion of our proposed framework with one last operation: tag release. In principle, we envision two ways that a tag can be released. The first one is realized by restoring the original identity of the tag. The second relates to transferring tag related secrets to the proxy of a new user. Below we explain how each case is handled.

In the first scenario, the user simply wants to bring a tag to its original state when for example she needs to return the tagged item for service. The manufacturer should be able to access the tag freely without the use of access control. One solution would be to use the mechanism described in Section IV-B and re-install the original identity of the tag. This, however, opens up the possibility of tag spoofing since a malicious proxy can re-implant a fake ID. So, we prefer a more direct approach where the tag itself *reverts* to the original state when instructed by the proxy, instead of relying on the identity *submitted* by the proxy. This is shown in Figure 4.

When the protocol is executed, the tag will revert back to its true identity, i.e. it will make $CID = TrueID$. Additionally, it may respond freely to scan requests. Inclusion of N_T in the third message is to make sure that this is a fresh request sent by someone who knows *both* the secret CID and K, i.e. the current owner. When, finally, the user reclaims back the tagged item, she can implant a new identity to prevent tag tracing.

In the second scenario, the user wishes to release the tagged item to a new owner/proxy. This is straightforward: the old owner gives the new one both CID and K. Then the new owner re-labels the tag with a new secret ID, thus ensuring both privacy and ownership of the tag.

V. Conclusions

In this work we have presented a framework for enhancing RFID security by means of a proxy, a personal device that assumes control of a user's tags. The proxy interacts with the tags but does more than simply simulating tags or acting as "device-in-the-middle" between tags and readers, encrypting reader queries and decrypting tag responses, as previous solutions do. To the best of our knowledge this is the first work that unifies past approaches and presents detailed protocols for such issues as tag acquisition, proxy authentication, resistance to privacy attacks, ease of transfer and release, and so on.

Once the proxy performs some initial transformations to the tags under its control, it can either *mediate* between tags and readers or let tags *directly* respond to scan requests. In the first case, the proxy can specify a number of policies that readers must comply with. In the latter, we make sure that tags do not emit any static identifiers thus providing ID anonymity and helping prevent tag tracing.

Overall, using the protocols described in the framework the user has full control of the tags she carries in a way that guarantees user's privacy and protection from a host of attacks like impersonation, cloning, tag spoofing and so on. However, we also make sure that our framework cannot be abused by malicious proxies. So, we guarantee that only authorized users can acquire and put tags under their control, user access to tags is authenticated, and when a user no longer needs the tagged product she can either make it readable to everybody or transfer it to another user in a way that guarantees the privacy of the new owner.

References

[1] Ari Juels, "RFID security and privacy: A research survey," In IEEE Journal on Selected Areas in Communication, 2006.

[2] M. Rieback, B. Crispo, and A. Tanenbaum, "RFID Guardian: A Battery-powered Mobile Device for RFID Privacy Management," in *Australasian Conference on informaiton Security and Privacy - ACISP 2005*, vol. 3574 of LNCS, pp. 184-194, July 2005.

[3] A. Juels, P. Syverson, and D. Bailey, "High-Power Proxies for Enhancing RFID Privacy and Utility," In *Center for High Assurance Computer Systems - CHACS 2005*, August 2005.

[4] Soo-Cheol Kim, Sang-Soo Yeo, Sung Kwon Kim, "MARP: Mobile Agent for RFID Privacy Protection," In *International Conference on Smart Card Research and Advanced Applications - Cardis*, April 2006.

[5] Nokia unveils RFID phone reader. In *RFID Journal*, 17 March 2004. Available at http://www.rfidjournal.com/article/view/834.

[6] H. Kato, K.T. Tan, "2D barcodes for mobile phones," in *Mobile Technology, Applications and Systems*, 2005

[7] A. Juels and R. Pappu, "Squealing Euros: Privacy protection in RFID-enabled banknotes," In R. Wright, editor, *Financial Cryptography 03*, pages 103121, Springer- Verlag, 2003. LNCS no. 2742.

[8] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for mixnets," In T. Okamoto, editor, *RSA Conference - Cryptographers Track (CTRSA)*, pages 163178, Springer-Verlag, 2004.

[9] Kenneth Fishkin and Sumit Roy, "Enhancing RFID privacy through antenna energy analysis," In *MIT RFID Privacy Workshop*, 2003.

[10] David Molnar and David Wagner, "Privacy and Security in Library RFID Issues, Practices, and Architectures," *Conference on Computer and Communication Security*, 2004.

[11] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm," In *Advances in Cryptology - Asiacrypt 2000*, Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed., Springer-Verlag, 2000.

[12] David Molnar, Andrea Soppera and David Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags," in *Selected Areas in Cryptography*, 2005.

[13] Tassos Dimitriou, "A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete," in *4th IEEE Intern. Conference on Pervasive Computer and Communications (PerCom)*, 2006.

[14] "Securing communications between mobile phones or other similar devices", SHA-1 fingerprint: 0x17503346d-69b83f1cc9c2c4a43ee748e250b29c4, MD5 fingerprint: 0xae8e0db-474913e9162e058521cae30a4, Version 2, Manuscript 2007.