TIME-MEMORY TRADE-OFF IN RFID SYSTEMS

Violeta Tomašević[1],
Milo Tomašević[2]

[1]Singidunum University,
32 Danijelova Street, Belgrade, Serbia,
[2]School of Electrical Engineering,
Bulever kralja Aleksandra 73,
Belgrade, Serbia

Abstract:
This paper elaborates on the application of Hellman's cryptanalytic time-memory trade-off approach in the RFID systems. First, the original method is briefly explained. Then, the situations and conditions that make the application of Hellman's idea possible in the RFID environment are discussed and analyzed. The effects of application of the time-memory trade-off method are illustrated for two RFID techniques: Ohkubo, Suzuki and Kinoshita's protocol, and constant-time identification protocol. For both of them the performance analysis is performed before and after application of the time-memory trade-off approach. Finally, the similarities and differences in modifications of the original method and its consequences in these two cases are pointed out.

Key words:
radio-frequency identification, time-memory trade-off,
OSK protocol, constant-time identification.

Acknowledgment:
This work has been partially supported by the Serbian Ministry of Education and Science (the projects III 44006 and III 44009).

1. INTRODUCTION

Radio frequency identification (RFID) technology enables the mass distant identification of objects (without any physical or visual contact) using radio waves. Along with the Internet, it is one of the most widely used technologies nowadays [1].

A lot of RFID protocols were proposed and developed [2-4]. Their efficiency heavily depends on the way of identification of tags associated with objects. Critical parameters of identification process are memory space and time needed for identification. Traditional contradiction between these opposing issues imposes the need for balancing between the two and calls for application of the time-memory trade-off (TMTO) approach in the RFID systems. This paper analyses the effects of applying TMTO methods in two RFID schemes: OSK (Ohkubo, Suzuki and Kinoshita) protocol and CTI (Constant-Time Identification) protocol.

Hellman's TMTO method is described in section II. General structure of an RFID system is given in section III accompanied by an analysis of possibilities to apply the TMTO approach in the RFID systems. Two examples of such an application of TMTO in the RFID schemes (OSK

Correspondence:
Violeta Tomašević

e-mail:
vtomasevic@singidunum.ac.rs

DOI: 10.15308/Sinteza-2016-124-130

and CTI) are demonstrated in section IV. Similarities and differences of these TMTO-based RFID schemes are discussed in section V. Finally, the conclusion gives a short summary of elaborations from the paper.

2. TMTO APPROACH

Time-memory trade-off is a general method for enhancing the performance of various kinds of algorithms. It was applied for the first time in the area of cryptanalysis by Hellman, who proposed a TMTO attack on DES block cipher in [5]. However, it occured that this method solves more general problem of inverting one-way function, so it can be successfully used in different occasions.

Using the TMTO approach problem of inverting function F: $x \rightarrow y$ is solved in two steps: pre-computation phase and online phase. By means of intensive calculations pre-computation phase collects the information about F, and stores them into the appropriate data structure. Online phase for a given y founds the corresponding x by using data structures from pre-computation phase.

Hellman's proposal is based on generation of m chains of fixed length k in pre-computation phase. The nodes of the chains cover the value space to which values x and y belong. The head node of the i-th chain is a randomly chosen value SP_i. The chain is generated by consecutive applying F function on value from the previous node, as demonstrated in Fig. 1. EP_i is the last node of the i-th chain. In order to save the space, only ordered pairs (SP_i, EP_i), $i = 1,...,m$, are saved.

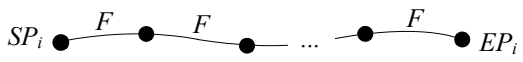

Fig. 1. Generation of the i-th chain

In the online phase starting with y value a chain is generated in the same way as in the pre-computation phase. For each new node a check is made whether it corresponds to some ending point EP_i. If not, a next node is generated. If an ending point is encountered, starting from corresponding SP_i from its stored pair, i-th chain is reconstructed until the node with value y is reached. The value from the node previous to y node is x we were looking for, since $y = F(x)$.

A problem in this procedure can happen because of false alarms. They arise because some value can have multiple occurences in the chains due to the fact that the output of F can be regarded as random. Consequently, it leads to merging of chains and results in irregular situations when either erroneous x is returned or x can not be found at all. A number of later proposals tried to overcome this problem of the Hellman's approach. One of the most significant improvements is based on *rainbow tables*, as described in [6].

3. TMTO IN RFID

With appropriate modifications Hellman's TMTO approach can be successfully applied in the area of radio-frequency identification (RFID).

RFID technology assumes three participants in the identification process: tag, reader and back-end server. The tag is a tiny device which contains a microchip with unique identifier stored into chip's memory and an antenna coil as a coupling element. It is inserted into an object which is to be identified. The reader communicates with the tags by radio waves (insecure channel), and with the back-end server using a secure channel. The back-end server has a database that contains information about the objects. It receives data from a reader and processes them (using own information) in order to identify the tag which sent the data.

RFID systems usually include very large number of tags, so the viability of their implementation practically depends on the unit cost of the tag. Necessity for a low-cost tag implies significant constraints on tag resources (predominantly memory space). It definitely guides some design decisions in choices/proposals of the protocols employed in RFID systems. The TMTO approach requires intensive calculations, so it can't be performed inside the tag. However, the back-end server has no such restrictions and can provide much more powerful resources. Rapid development of computer technology brings a constant enhancement of available resources. Consequently, some protocols are gaining the importance as time goes by *e.g.*, improved performance of state-of-the-art hardware can enable an increased number of tags that a system can support or speeding up the identification process of a tag. Also, it can enable the viability of some already proposed protocols practically infeasible at earlier stages of technology.

In an RFID system, TMTO approach can be used in the process of tag identification on the back-end server. The primary goal is to achieve the appropriate trade-off between available memory space and acceptable

tag identification latency. For a given back-end server configuration, we can establish the relations between parameters of importance (number of tags, number of tag accesses, amount of memory, precomputation time overhead, on-line identification latency) and determine its limits for which an RFID system can operate regularly. By adjusting these parameters one can influence the functioning of an RFID system.

4. CASE STUDIES

This section elaborates on two examples of the TMTO-based RFID systems: OSK (Ohkubo, Suzuki, Kinoshita's) protocol and CTI (Constant-Time Identification) protocol. First, the original versions of the OSK and the CTI protocols are explained, and their TMTO-based enhancements are analysed afterwards.

OSK protocol

OSK belongs to the class of hash-chain protocols and it is described in [7]. In the preparation phase, every tag in the RFID system is initialized with two randomly chosen values. The identifier ID_i and inital state s_i^1 are associated with the i-th tag. State s_i^1 is stored into tag's memory, while the ordered pair (ID_i, s_i^1) is stored into database of the back-end server.

The protocol employs two hash functions, H i G, which can be executed both in the tag and on the back-end server. When i-th tag is queried by the reader, this tag generates $G(s_i^k)$, where s_i^k represents current tag state, and k is number of accesses to this tag until that time. The value generated in this way is sent to the reader. After that, the tag modifies its own state to value $s_i^{k+1} = H(s_i^k)$. Since the tag changes its state on each access, the set of subsequent tag states can be represented by a hash chain, where the value of the next node is obtained by applying H function on the value of the preceeding node.

After the reader receives some value it forwards this value to the back-end server which starts the identification of the tag that sent the value. In the process of identification, the back-end server generates hash chains by consecutive applying H function starting from tag's initial values s_i^1 (head nodes of the chains). The number of the nodes in a chain is m which also represents maximum number of the tag queries. The number of chains is equa to the number of tags. For each node in a chain G function is applied and obtained hash value is checked on equality with the value received from the reader. If it is not equal, the next node is generated. If the current chain is exhausted, the next chain is started. If equality is found, then the head node of the current chain is the initial state of the tag that should be identified and its identifier can be easily found. The principal functioning of the OSK protocol is illustrated in Fig. 2. In this figure, comparator for matching the obtained value and result of G function is denoted by CMP.

For an RFID system with n tags, described identification procedure requires nm calculations (executions of hash functions, G or H). Because of its high computing complexity, the OSK protocol can be practically used only in systems with relatively small number of accesses (*i.e.*, $n = 2^{20}$, $m = 2^7$).

A significant enhancement of the OSK protocol (OSK/AO) was proposed by Avoine, Dysli and Oechslin in [8]. It was the first case that TMTO strategy is applied in the RFID area. An improvement is attained by more efficient procedure of tag identification on the back-end server. Following the TMTO approach, the pre-computation phase is introduced. During this phase, the OSK/AO protocol generates the chains whose nodes correspond to hash values that the tags can send. Hash values in a chain are intermixed and can be sent by different tags. The first and the last node of each chain is stored into a table. In the identification process, these stored data make the searching of tag state at the time it was queried by the reader more efficient. Unlike brute force approach of the original OSK protocol where on average a half of all chains are searched, the tag state in the OSK/AO protocol is found by reconstruction of only one chain.

The initialization of the RFID system with OSK/AO is performed in the same way as with the OSK protocol. Also, an additional overhead is incurred by generation of the chains with outputs of G function which the tags can send. It is considered that G function outputs the random values, so it isn't possible to produce next random hash value from a previous one directly during chain generation. Therefore, it was necessary to employ a reduction function R. From some random value this function produces a pair (p, q), where p is interpreted as an ordinal number of the tag, and q is the current number of accesses to the p-th tag. With (p, q) a value is calculated as $G(H^{q-1}(s_1^p))$, which represents one of the regular values that p-th tag can send. This value is assigned to the next node in a chain. This procedure is illustrated in Fig 3.

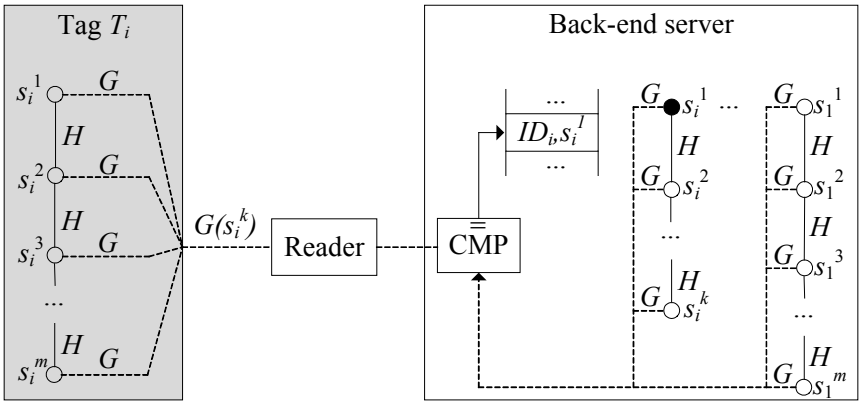

Fig. 2. Principal functioning of the OSK protocol

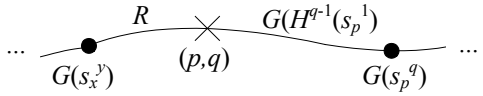

Fig. 3. Generation of the next node in a chain in the OSK/AO protocol

According to this procedure, the chains contain only hash values that the tags can regularly send, but in an arbitrary order. After a certain number of chains of predetermined length are generated, only the values from the first and last node in each chain are memorized.

After the OSK/AO-based RFID system is initialized, its three components communicate in the same way as in its OSK-based counterpart. After each query, the tag modifies its state as explained previously.

When a hash value is received by the back-end server, the identification process for the tag which sent this value is started. Identification assumes the generation of an online chain whose starting node corresponds to the received value. Each subsequent node in the chain is generated in exactly the same way as in pre-computation phase illustrated in Fig 3. For each new node a check is made whether it corresponds to the ending node of any chain (searching the table stored in the initialization phase). If not, the next node is generated. If the end of some chain is encountered, the corresponding starting value is read from the found stored pair. Then, the chain is reconstructed until the node with the received this hash value is reached. By applying the *R* function on the value from the previous node, a pair of integers is obtained and the first element of this pair represents the ordinal number of the tag which sent the value. In this way, the identification is successfully completed.

Since the output of the *R* function is a random pair (p, q), there is a possibility that the same hash value appears in the chain nodes more than once. It may result in identification of a wrong tag. Also, because of the restricted dimensions of the chains, it may happen that some hash value is not embedded in the chains at all, so the identification is impossible. In order to decrease the number of these irregular situations, in the initialization phase the OSK/AO uses the *rainbow tables* method [6].

The complexity of the OSK/AO algorithm in the preparation phase can be approximated to $nm^2/2$. Comparing to the OSK, an additional memory space is needed for storing the chain starting and ending points. However, complexity of the online identification in the OSK/AO is much lower since it can be calculated as a product of chain length and $m/2$, while it was mn in the OSK. Finally, an increased complexity of initialization phase is amortized by more intensive accessing of the tags.

CTI protocol

Just like the OSK, the CTI is a hash-based RFID protocol [9]. The basic idea of the CTI protocol is to achieve constant identification time for each tag in an RFID system at the expense of larger amount of memory on the back-end server. It is an entirely different approach from the OSK which has linear time identification complexity.

In a CTI-based RFID system, every tag has a secret key *k*, a secret pseudonym *d* and an internal counter *c*. The key and pseudonym are updated after each successful tag-reader mutual authentication, while the counter is incremented after each authentication (successful or

unsuccessful). When the counter reaches its maximum value C, it is reset to 0. The number of tags in the system is N_T, and the number of pseudonyms is $N > N_T$.

In the initialization phase, each tag is assigned with two random values k and d, while c is reset to 0. Also, a database with information neccessary for identification is built on the back-end server.

Functioning of the CTI protocol is illustrated in Fig. 4.

First, the reader addresses a tag by sending a random nonce r to it. The tag uses hash function h and calculates two values, $h(d,c)$ and $h(0,d,c,k,r)$, returns them to the reader which forwards the first value to the back-end server. Database on the server consists of three tables. The first table has 2^n entries, where n is the length of the truncated hash value $h(d,c)$. Each entry of this table contains a pointer to a mini-table inside the second table in the database. Each mini-table stores hash values $h(d,c)$ with the same position, *i.e.* the same n most significant bits. The second table with NC entries can store all the possible hash values that the tags can send. In addition to hash values, each entry is assigned with a pointer to an entry of the third table which stores the information about the tags corresponding to the given pseudonym. Each entry also contains the values c and d. The third table has N entries.

When a hash value M is received by the server, the identification process starts with extracting its n most significant bits. The extracted value (here dentoted as M^n) is used to address an entry of the first table. It further points to the appropriate mini-table containing hash values with the n most significant bits that correspond to M^n. If the mini-table has more entries, it is searched for the value M. When such an entry is found, its content is read and used as a pointer to the field of the third table which contains the data about tag currently associated to the pseudonym (secret key and identifier of the tag).

After that, a new pseudonym d' is selected from the pool of unused pseudonyms and it is sent to the reader, along with k and d. Then, the reader forwards these data to the tag in a way indicated by Fig. 4 preserving the authentication of the reader, as well as secure transfer and integrity check of the new pseudonym. After changing the tag's pseudonym, the pointers into the database must be updated. When the authentication of the reader is done, the tag sets its pseudonym to d', secret key to $k'=h(k)$, and c to 0.

Because of the multiple level of hashing, the CTI protocol is very time efficient. However, excessive large demands on memory makes its practical use questionable. For example, an RFID system with 10^9 tags, twice as more pseudonyms and 10^3 tag accesses demands the memory of 31TB (predominantly for the second table of the database - 19TB) [10].

In order to decrease the memory demands for implementation of the CTI protocol, in [11] Chang *et al.* proposed the modification of the original protocol by application of the time-memory trade-off approach in initialization of the database. The result is the modified CTI protocol named the CTI/TM. This protocol defines the function $F(i,j) = h(d_i,j)$ and the reduction function $R_t(x) = (i',j')$, where i' and j' depend on t and x. It can be regarded that these functions return the random values. The TMTO chains are generated by consecutive application of F and R_t functions on each node. The method of *rainbow tables* [6] is also used during this procedure. Finally, only the starting and ending nodes of the chains are stored into database.

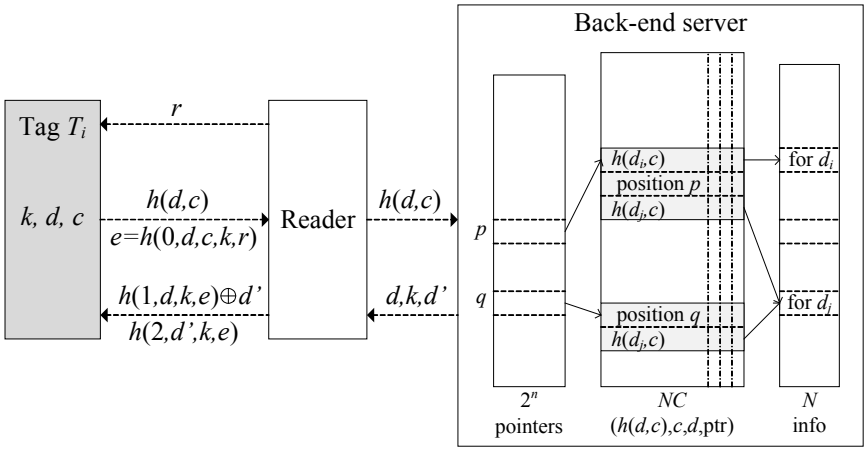

Fig. 4. Principal functioning of the CTI protocol

Besides the modified structure of the database on the back-end server, the CTI/TM implies some changes on the tags, too. Now, along with the h function each tag can use g hash function to increase randomness and decrease the probability of collisions.

According to the CTI/TM protocol, the communication starts when the reader addresses the tag by sending some random value r. Based on its current state, the tag calculates the following hash values: $m=g(c,k)$, $h(d,m)$ and $r'=h(0,d,m,k,r)$. Then, the counter c is incremented and two last hash values are sent to the reader which forwards them to the back-end server. Identification is performed by searching for the received value $h(d,m)$ in the stored chains in order to find the pair (d',c') which produced it. With the obtained pseudonym d', the key value k is read, and then an equality $r'=h(0,d',c',k,r)$ is checked. If it holds, the tag is properly identified and the subsequent procedure is the same as in the CTI protocol.

The CTI/TM protocol significantly reduces the memory demands. It can be shown that memory can be saved up to 89,5% in relation to the original CTI, still having large probability of successful identification.

5. DISCUSSION

As it can be concluded from the previous elaboration, the application of the Hellman's time-memory trade-off approach in RFID systems requires some modifications of the original protocol and its adaptation to a particular scheme. Although the OSK and the CTI schemes are quite different, the contexts of applying the TMTO method are very similar. Namely, in both cases, the chains store the hash values that the tags can send. Since these values can be regarded as random, its direct chaining (in the sense that one directly produces the other) is not possible. On the other hand, because of an enormous value space, the chains can not include all the values. A part of possible hash values that could be sent by the tags can be chained only by introduction of the reduction functions. During generation of each new node in the chain, the reduction function randomly selects the number of the tag and the ordinal number of the query for which the appropriate hash value is calculated. This procedure is probabilistic since it can happen that the same hash value appears in the chains more than once. Luckily, the probability of multiple occurence is insignificant (below 0,1%) because the tags use only a small part of value space.

Regardless of some similarities in applying the TMTO approach in to the OSK and the CTI protocols, the achieved effects are qualitatively different. In case of the OSK, the TMTO approach led to the increase of the memory demands and decrease of the time overhead. Additional memory is spent for storing the starting and ending points of the chains, which enables faster tag identification by searching only one chain. In the CTI method, the TMTO approach exhibits the opposite effect. The memory demands are significantly reduced at the expense of increasing the tag identification time. The chaining of hash values from the second table on the back-end server and storing only starting and ending points of the chains saves considerable memory. However, identification was slowed down since the direct access to the tables is replaced by searching the chains.

6. CONCLUSION

It was demonstrated that cryptanalytical TMTO method can be effectively combined with the RFID technology for the sake of improving the efficiency and practical feasibility of an RFID protocol. Capabilities of the TMTO involvement are illustrated by the examples of two hash-based RFID schemes, the OSK and the CTI. In both cases, enhancements are achieved due to the chaining hash values that could be sent by the tags. Although these RFID schemes are the only examples of TMTO application in the RFID area, it is evident that the TMTO approach is quite flexible in balancing the time and memory complexity of the system for this specific purpose. Consequently, further research in applying TMTO in some other existing RFID system seems to be quite appealing.

REFERENCES

[1] M. Kaur, M. Sandhu, N. Mohan, and P. Sandhu, "RFID Technology Principles, Advantages, Limitations & Its Applications", Int. Journal of Computer and Electrical Engineering, vol. 3, no. 1, pp. 151-157, 2011.

[2] A.Juels, and S.Weis, „Defining Strong Privacy for RFID", IEEE Int. Conf. on Pervasive Computing and Communications, pp. 342-347, 2007.

[3] J. Wu, and D. Stinsion, "A Highly Scalable RFID Authentication Protocol", Information Security and Privacy, vol. 5594 of LNCS, pp. 360-376, 2009.

[4] G. Avoine, A. Bingol, X. Carpent, and S. Yalcin, "Privacy-fiendly Authentication in RFID Systems: On Sub-linear Procols based on Symmetric-key Cryptography", IEEE Transactions on Mobile Computing, Issue 99, 2012.

[5] M. Hellman, "A Cryptanalytic Time-Memory Trade-Off", IEEE Transactions on Information Theory, IT-26, no. 4, pp. 401-406, 1980.

[6] P. Oechslin, "Making a Faster Cryptanalytic Time-Memory Trade-Off", Advances in Cryptology, vol. 2729 of LNCS, pp. 617-630, 2003.

[7] M. Ohkubo, K.Suzuki, and S. Kinoshita, "Cryptographic Approach to "Privacy Friendly" Tags" RFID Privacy Workshop, MIT, USA, 2003.

[8] G. Avoine, E. Dysli, and P. Oechslin, "Reducing Time Complexity in RFID Systems", Selected Areas in Cryptography, vol. 2897 of LNCS, pp. 291-306, 2005.

[9] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable RFID systems: a privacy-preserving protocol with constant-time identification", Parallel and distributed Systema, IEEE Transactions on 23.8, pp. 1536-1550, 2012.

[10] T. Martin, "Privacy in RFID systems", PhD Thesis, Universite catholique de Louvain, Belgium, 2013.

[11] J. Chang, H. Wu, and D. Zhang, "On Constant-Time-Identification and Privacy-Preserving RFID Protocols: Trade-Off between Time and Memory", IEEE Int. Conf. on Ubiquitous Intelligence & Computing and IEEE Int. Conf. on Autonomic & Trusted Computing, 2013.