Ad Hoc Networks 11 (2013) 1619–1628

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Lightweight protocols and privacy for all-in-silicon objects

Denis Trček *

Laboratory of e-media, Faculty of Computer and Information Science, University of Ljubljana, Tržaška c. 25, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 24 May 2012
Received in revised form 13 February 2013
Accepted 27 February 2013
Available online 14 March 2013

Keywords:
Wireless sensor networks
RFID technology
Security
Privacy
Lightweight protocols
Security policies

ABSTRACT

Pervasive computing is already becoming a reality and one crucial consequence of this fact is endangered privacy. Now taking into account typical properties of pervasive computing devices, which are weak computing power and stringent energy or power consumption limitations, lightweight solutions are a must. This especially holds true for all-in-silicon objects like radio frequency identification tags, or RFIDs. Many solutions in this area are called lightweight, but being lightweight requires conformance to quantitative requirements using certain metrics. A solution that adheres to such requirements is a new privacy enabling protocol for RFIDs that outperforms other architecturally similar protocols, and this presents the first contribution of this paper. Further, privacy is not only a matter of technical solutions, but increasingly so a matter of organizational processes. This fact calls for further addressing of supporting its formal treatment in business contexts. This paper provides a basis for formal addressing of privacy from business processes perspective, and this is its second main contribution.

1. Introduction

Internet of things (or IoT) is at the core of pervasive computing, and it can be defined as "wireless self-configuring network between objects" [1]. This definition is technology oriented, while a more business processes oriented one goes as follows: "IoT is a world where physical objects are seamlessly integrated into the information network, and where these objects can become active participants in business processes. Services are available to interact with these 'smart objects' over the Internet, query and change their state and any information associated with them, taking into account security and privacy issues [2]".

IoT is therefore addressing a wide variety of computing devices, which share key typical properties: limited computing resources, limited energy/power autonomy (or even none), and wireless connection to the global internet. During recent years IoT resulted in some typical families ranging from the most "primitive" all-in-silicon devices like radio frequency identification tags (or RFIDs) on one side, and IP protocol enabled wireless sensors, supported by microcontrollers, on the other.

Therefore IoT world is quite diverse, but without doubt, RFID objects are the most demanding ones when it comes to efficient protocols. So if a protocol is suitable for implementation in such objects, it would make sense to refer to it as a lightweight one. And taking further into account that this kind of devices will be one of the most numerous devices that we will be interacting with, privacy issues get to the forefront. Technological development in this area is already under strong legal pressure [3]: In the European Union the alma mater of related legislation is EU Data Privacy Directive [4], while in the US this legislation is more fragmented, but is gaining momentum – one such example is California, which has enacted a legislation related to privacy and RFID technology [5].

This paper presents two important contributions in the area of privacy. It first focuses on a holistic model of privacy from a business perspective – despite many efforts in this domain, adequate formalization of privacy for security policies still needs inputs. The reason is that privacy has been formally covered so far almost exclusively at

* Tel.: +386 1 4768 918; fax: +386 1 4264 647
E-mail address: denis.trcek@fri.uni-lj.si
URL: http://www.fri.uni-lj.si/en/laboratories/lem/

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.02.005

the technical level. Consequently, the gap toward upper levels of organizational and business processes remains an open issue. And this paper presents a solution to fill this gap by deploying Petri nets. Further, a new technical privacy enabling solution for IoT is presented. This is a non-deterministic privacy enabling (ND-PEP) protocol that takes harsh technological reality into account. It is lightweight and implementable in RFIDs within the well-known "5 cents price limit". It should be added that although the notion of lightweight protocol is widely used, it has rarely some formal grounds – a metric that enables measuring how lightweight a protocol will be taken into account in this paper. It uses the number of required NAND gates for a protocol implementation as a metric, and it will be taken into account also in this paper.

The paper is structured as follows. In the second section privacy in business contexts (and its formal treatment in relation to security policies) is addressed. In the third section a new, non-deterministic privacy enabling protocol is presented and analyzed, being preceded by some background that is needed to understand the harsh technological reality that has to be taken into account in such solutions. There are conclusions in the fourth section, followed by appendix with a formal representation of the new protocol, while the paper ends with acknowledgments and references.

2. IoT and privacy – from technology to business processes

The first front in verification of privacy (and security solutions in general) is at the technical level, where formal methods play a central role. Once formally verified, these solutions become implemented and often deployed in organizational and business environments. Consequently, addressing privacy – and security in general – requires business processes perspective, because IoT devices are becoming their integral parts. And this kind of addressing is embodied in security policies (the main standards family in this area is ISO 27000 family [6]).

Now getting to privacy formalization that includes business processes views, surprisingly little research in this area exists. One recent exception is [7], where privacy is formalized in a form of enforced purpose restrictions to achieve adherence with privacy policies. This approach is based on Markov decision processes and is addressing privacy implicitly at the business processes level. More precisely, it is assumed that privacy is somehow properly covered by security policy, while this formalism then serves to verify the compliance of the observed procedure with this policy.

To provide explicit treatment of privacy in business processes context, the starting point is a definition of privacy. There exist many definitions of privacy, but we will refer to

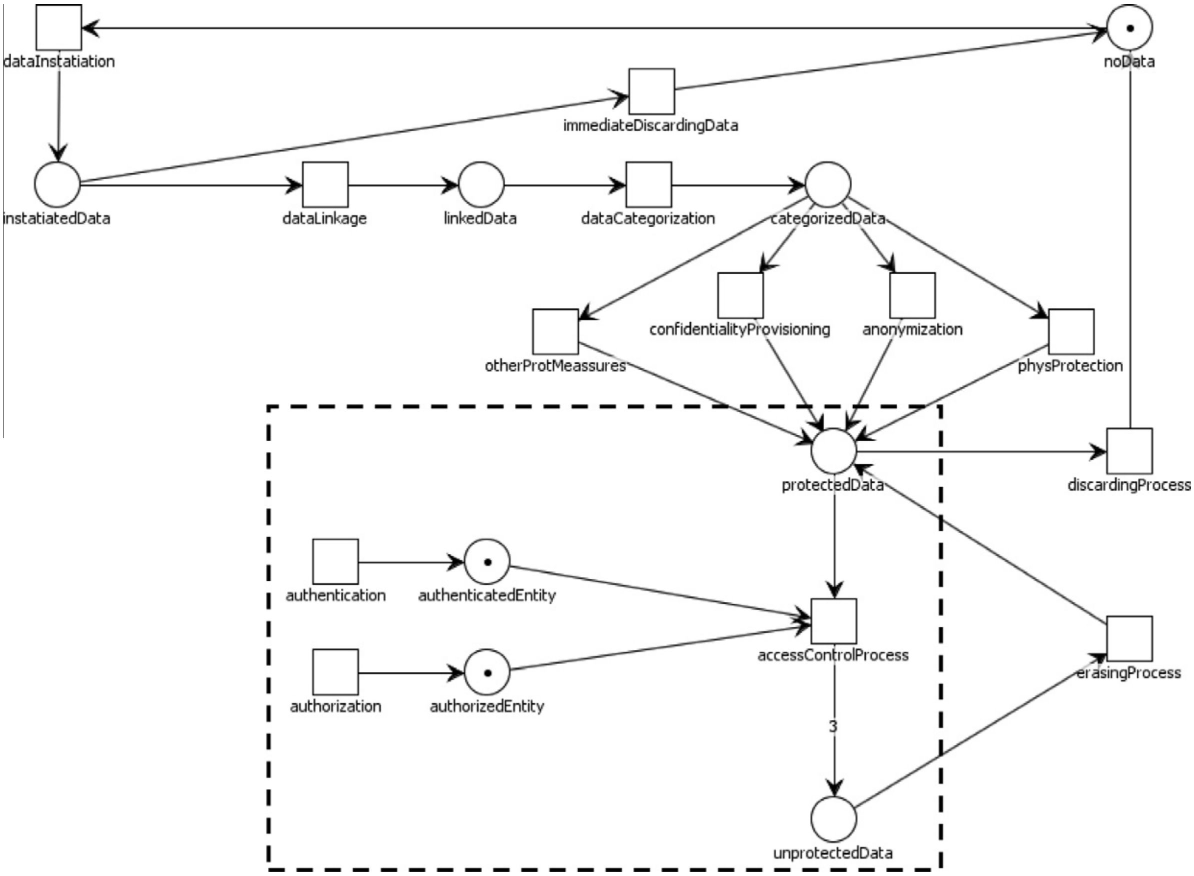

Fig. 1. High level privacy model (the dashed line denotes its technological part).

most authoritative sources. The first is Merriam-Webster dictionary, which states that privacy is "the quality or state of being apart from company or observation" and "freedom from unauthorized intrusion". The second source is one main legislation act in this area, the EU Data Privacy Directive. It states about privacy, more precisely, protection of personal data, the following: "Personal data are any information relating to an identified or identifiable person." This covers direct and indirect identification by referencing factors that include physical, physiological, mental, economic, cultural or social identity. Personal data, once identified as such, are afterward subject to various (legislation based) protection procedures about their storage and transmission. A closer look at these definitions reveals two conceptually different views. While the first definition is very narrow and focused on the level of an individual, the second one extends it to procedural level that covers business processes in organizations, and related processes in societies in general. From this second one it follows that the first step is to recognize private data as such and to categorize them accordingly. Next, these data have to be protected. Afterward, the protected data may be disclosed (manipulated) only in line with legally acceptable procedures. In the end, after a legally defined period, data have to be destroyed.

This high-level, procedural reasoning can be formally depicted by Petri net model that is given in Fig. 1. Within this high-level model, a technology focused (narrow) definition of privacy can be obtained (see the dashed-line rectangular area in Fig. 1). This definition goes as follows: Data privacy is a composite security service that consists of the three basic security services, which are confidentiality, authentication, and access control, and where the assumed initial service is confidentiality.

It should be quite straightforward to deploy the above model and the derived definition as a basis for informal verification of security policies. Moreover, also formal verification is quite straightforward – having privacy defined with security services [8], many established formal methods for verification of these security services can be used, e.g., BAN logic [9] or SvO logic [10] for authentication, soft constrained programming for confidentiality [11] (note that protected data in our case equal to confidential data), and specialized calculus for access control [12].

3. Non-deterministic lightweight privacy enabling protocol

This section gives a new protocol that extends a family of two protocols for privacy provisioning in RFID environments [13]. In order to better understand its specifics, the technological reality of IoT has to be given first.

3.1. Technological reality of IoT and privacy issues

This sub-section provides a taxonomy of IoT devices that is focused on their technological determinants:

- At the lowest level there are the most primitive objects that are implemented "all-in-silicon", being unable to run software applications. This category includes bare sensors and RFIDs:
  - RFIDs consist of numerous clocked NAND gates that perform various logic functions on inputs and output the results. However, in the most basic case RFIDs only respond by providing some static value like electronic product code, EPC. They are typically passive and powered by a reader. They contain up to 1 KB memory, while their communication depends on air-interface, providing only physical and link layer capabilities [14]. Also near field communication, NFC [15], belongs to this category, because it is a kind of successor of RFIDs.
  - Bare sensors consist of analog circuits that acquire environmental data, and NAND gates that perform analog to digital conversion. Once in a digital form, more or less similar circuitry as that of air-interface for RFIDs can be used for communication. In cases where sensors are directly physically linked, two kinds of buses are used, I^{2}C [16] and SPI [17], which require additional logic gates.
- At the next level there are still "all-in-silicon" structures, but actively powered: RFID structures, sensor structures, actuators, and their combinations, i.e., hybrids. These structures induce approx. two orders of magnitude higher costs than passive devices, because of their dependence on batteries [18]. Their typical advantage is not only increased communication range, but also computational capabilities (in case of actively

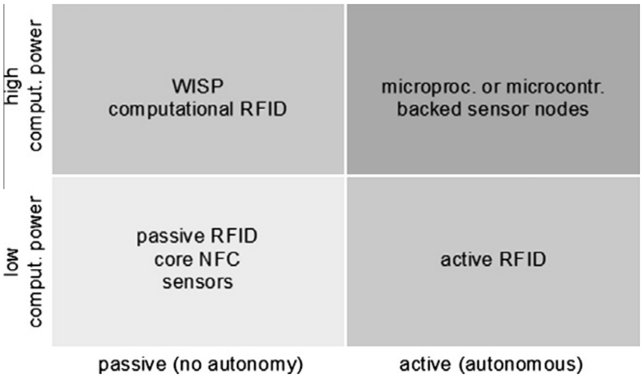

Fig. 2. Two-dimensional taxonomy of IoT devices.

powered RFIDs, primary memory could go up to 128 KB already some 5 years ago [18]).
- At the next level there are "all-in-silicon" devices backed by microcontrollers or microprocessors, but passively powered. In case of RFIDs, these are referred to as computational RFIDs, CRFIDs [19]. They have typically up to 8 KB of flash memory and 256 bytes of RAM, but they still depend on air-interface. A notable architecture here is Wireless Identification and Sensing Platform, or WISP [20]. This platform enhances RFID functionality with sensing (quantities include light, temperature, acceleration, strain) and computing, while still harvesting the power from radio signals emitted by the reader.
- At the highest level there are autonomously powered IoT devices that are backed by micro-controllers. Examples of such devices are numerous, with probably the most typical representative being now mobile phones, where energy limitations are not so stringent anymore, nor computational resources. Therefore sophisticated networking can be provided at physical and link layers like IEEE 802.15.4 [21], or even IP networking, where an adapted standard for this kind of devices, 6LoWPAN, exists [22].

The above structuring suggests that one-dimensional taxonomy of IoT devices is not sufficient. To make a sensible taxonomy, these devices have to be categorized at least with respect to energy (power) autonomy and computational power (see Fig. 2).

The focus of this paper is the first population mentioned above. More precisely, pure RFIDs will be covered, which are in principle limited to all-in-silicon implementations, are passively powered, and have no microcontroller or microprocessor support. They have in common also communication frequency bands that are at 13.56 MHz and between 860 MHz and 960 MHz (as to these frequencies we anticipate that RFIDs will also benefit from the latest research in the area of cognitive radio [23]). So this population, due to its limitations and stringent requirements, remains most challenging for provision of privacy.

3.2. Quantitative metrics for lightweight protocols

The technological reality of RFIDs is bound to NAND gates. This does not limit the number of implementable Boolean functions, because any logic function can be implemented if a so called complete set of logic functions is available; and one such set consists of negation and conjunction. Implementing it with NAND gates requires one NAND gate for negation, and two NAND gates for conjunction. Further, RFIDs require clocked gates, and this gives a basis for counting the required number of NAND gates to implement the typical basic crypto-mechanisms building blocks [24]:

- For storage D memory cells are used, and each such cell requires five NAND gates.
- For bitwise XOR operation four NAND gates are required for each pair of bits.
- One-bit full adder can be implemented with eleven NAND gates.
- Addition modulo 2^n requires n-times eleven NAND gates.
- Pseudo-random values can be generated with shift registers (the mathematics behind this approach is explained in [25]). A shift register with four shifting bits requires approximately 60 NAND gates, with 8 bits approximately 120 gates, and so on. More promising solution that is supposed to produce random numbers by deploying digital circuit artifacts is given in [26], but it requires about 300 NAND gates.
- Light DES, DESL, which is a block cipher that is used for symmetric encryption and hashing, requires approx. 1800 gates [27]. DESL encrypts 64 bits of plaintext in 144 clock cycles and will be the basis for producing 96 bits long hashed values (therefore two keys can be used for hashing and the result can be truncated). Ordinary hash functions like SHA are inappropriate, because their implementation in hardware would significantly exceed the number of logic gates needed for DESL (various principles of using symmetric block ciphers for one-way hash functions can be found in [28]). Alternatively, one could use AES implementation, which requires approx. 3000 NAND gates [29].

Despite Moore's law (which in principle still holds true for RFIDs) it is sensible not to count on more than approx. 7000 logic gates for security. Already in 2006 it was assumed that approx. 2000 gates could be allocated to security within economically acceptable range, and taking into account Moore's law the currently accepted number would be around 16.000 gates. But due to market pressure, where the five cents limit for RFIDs is pushed down, a reasonable estimate for the ceiling number of security dedicated gates today would be somewhere between 6000 in 8000 logic gates.

Finally, according to metrics given in [30], to measure how lightweight a protocol is, the total number of NAND gates for storage, logic functions and data transfer is taken (from a tag's perspective). Data transfer cost is measured in storage cells needed to contain the communicated messages.

3.3. Existing solutions and their analysis

To present the main contributions of the new protocol, some typical most relevant solutions proposed so far will be analyzed (an extensive and most up-to date review of the literature in this field can be found in [31]):

1. Ohkubo, Suzuki and Kinoshita have proposed a protocol where a tag and a reader share an initial secret s_i and two hash functions G and H [32]. After being triggered by the reader, the tag computes $H^1(s_i) = H(s_i)$, which presents a new secret and is stored. At the same time the value $G^1(s_i) = G(s_i)$ is computed, and this is sent to the reader, where each of the stored values is hashed to find the match $(ID, G^1(s_i))$. When being interrogated for the next time, the tag computes $H^2(s_i) = H(H(s_i))$,

while the reader receives $G^2(s_i) = G(H^1(s_i))$ and searches for the match (ID, $G^2(s_i)$) to identify the tag.

2. Henrici and Muller have proposed a protocol where a tag and a reader share a common hash function H [33]. After being interrogated by the reader with r, the tag calculates $H(ID)$, $H(s\bullet ID)$ and δ_s, and sends this triplet to the reader (ID is tag's identifier, s, is the number of the current session, δ_s is the difference between the current and the previous session number that serves to stay in synchronization with the database, and "•" denotes some chosen operator). When the reader receives the triplet, it computes a new $ID_{new} \leftarrow ID\bullet r$ for the tag and sends back r and $H(r\bullet s\bullet IDnew)$. Upon receipt, the tag is able to check the integrity of r and is able to calculate the ID_{new}.
3. Weis, Sarma, Rivest and Engels have proposed a protocol where a reader and a tag share a secret s and a hash function H [34]. After being interrogated by the reader, the tag computes a pseudo-random r and computes (r, ($H(ID||r)$), where "||" denotes concatenation. Upon receipt, the reader starts verification. If the checks are successful, it replies with the tag's ID.
4. Some less frequent approaches deploy even public-key cryptography [35], where authors have proposed access control of a reader on the basis of its authentication, verifiable location of the reader during interrogation, and verifiable time of interrogation. The main idea of the proposal is using public key cryptography that is deployed in a way where the public key of the reader remains secret (and is embedded in a tag). This enables better scaling than with symmetric key based approaches, where numerous tags share the same key with the reader and therefore the reader has to perform extensive number of verifications to find a match in a database that reveals tag's ID.

An analysis of the above protocols follows – it addresses security weaknesses and evaluates them with the metric mentioned above:

- As to the first protocol, reply attacks are possible, because the second message sent from the reader is not linked to the first message. To cure this issue, the first message should contain a fresh challenge r, while the form of the second message should be $G(si\oplus r)$ [36]. Further (and we are not aware of any research paper reporting this), the protocol still remains vulnerable to de-synchronization attacks. If an attacker simply blocks one or more responses from the tag, the authorized database will not be able to track the tag due to expecting a particular value from a hashed ID chain, but receiving a value with some different sequence number. Now as to the number of required NAND gates for the corrected version, the tag has to be able to store ID and its two hashed outputs, which require $2*5*96$ NAND gates (assuming lengths of 96 bits). Next, it has to calculate two hash functions where we will assume a symmetric cipher DESL with two different keys, so the required number of gates is approx. 1800, while for the two keys it is $2*64*5$ gates. Finally, XOR-ing is needed between s_i and r requiring $96*4$ NAND gates. Therefore the total number of required gates for storage and logic functions is approx. $960 + 1800 + 640 + 384 = 3784$. Now as to the gates equivalent required for communications, the first message can be neglected, while the second massage requires $2*96*5$ gates, so the total cost of the protocol is approx. $3784 + 960 = 4740$ gates.
- As to the second protocol, the operation denoted by "•" can be XOR operation. So if an attacker sends the message $H(r\oplus\ s\oplus IDnew) = H(s\oplus IDnew)$ in the third step, where r is a zero-bit sequence (it is read in plain in the second message), the produced message is the same as that from the second step. Thus the tag updates its ID with a value that differs from the one stored in the database and the system falls out of synchronization [37]. As to the number of required gates, for hashing we will again assume DESL, which requires approx. 1800 gates. Next, storage is needed for the secret ID; assuming again its length is 96 bits, the required number of NAND gates for its storage is $96*5$. Further, XOR-ing has to be performed, which requires $96*4$ NAND gates. Next, two hash values in step 2 (the same locations can be used in step 3 for r value and one hash value) have to be stored, which requires 2 memory locations, i.e., $2*96*5$ gates. Finally, in step 3 hashed values have to be compared, which requires additional $2*96*5$ gates for storage of the received and calculated values that are compared afterward. Ignoring now the gates needed for bit by bit comparisons, the total number of gates for storage and logic functions is $1800 + 480 + 384 + 960 + 960 = 4584$ gates (δ_s related gates have been ignored). As to the cost for communications and neglecting the first message, the cost is approx. $2*96*5 + 2*96*5 = 1920$ (δ_s related gates have been ignored again). The total cost is therefore approx. 6500 gates.
- With the third protocol, the first problem is sending the last message in plain. ID is the core of the game and traverses the medium unencrypted, which violates the basic principles of designing such protocols [38]. Further, the first message is not cryptographically linked to the third message (again, we are not aware of any research paper reporting this weakness, although we have found no attack so far). As to the number of required gates, for hashing we will again assume DESL, which means that approx. 1800 NAND gates are needed. Next, storage is needed for the secret ID and the random value r; assuming both lengths are 96 bits, the required number of NAND gates for its storage is $2*96*5$. Next, random r is calculated by the tag, and assuming 96 random bits, the number of required NAND gates is $24*60$ gates. Therefore the total number of gates for storage and logic functions is approx. $1800 + 2*96*5 + 24*60 = 4200$. Calculating communication costs, where we neglect the first message, we get $2*96*5 = 960$ gates, so the total cost of the protocol is approx. 5160 gates.
- As to the fourth solution, putting crypto-protocol attacks aside and focusing on how lightweight this protocol is, already the core number of required gates, as stated by the authors, is 3.300 for public key encryption, and 159 bytes for system memory, which means $159*8*5$ gates, i.e., 6.360 gates (the last factor in this

multiplication is a consequence of technological reality, where storage of one bit with clocked NAND gates requires 5 such gates – more details will be given in the fourth section). Further, gates are also needed to implement the protocol automaton (four transitions and exchange of related messages), not to mention that the protocol depends on reporting location (from a near-by GPS), and interfacing an RFID to GPS requires some bus, which comes at additional cost in terms of logic gates needed. As a result, even if we ignore communication costs this solution requires well over 10.000 logic gates and exceeds lightweight protocol architectures considered in this paper. Therefore although the approach is interesting and provides some advancement, its deployment in operational environments is not realistic for some years to come.

3.4. Non-deterministic privacy enabling protocol – ND-PEP

The main idea around which the new protocol is built is deployment of computational power asymmetry between a reader and a tag. This is the basis for using non-determinism, which means that responses from a tag are randomly distributed among equally possible values. And while a tag can calculate a random value at a low cost, the checking on the other side requires more power, because the reader knows only the interval where these random values could be coming from, so it has to check all of them against those in a database. Once a match is found, the procedure is successfully completed and the tag is authenticated.

The new protocol goes as follows (see Fig. 3). A reader and a tag are both able to compute the same strong one-way hash function H. The tag and the reader also share a value k that determines the number of random bits that are used in the process of using aliases that provide privacy.

Now when a user with a tag comes within the range of a reader, the tag is challenged to authenticate itself towards the reader. However, as the reader may be an adversary that collects privacy related information, the corresponding procedure goes as follows:

1. The reader challenges the tag with a random value.
2. Upon receipt of the challenge, the tag optionally checks it against the set of the last m received challenges to ensure that the challenge is fresh. These challenges are stored in a first-in-first-out (FIFO) memory. If the received value is fresh, it enters the FIFO memory and the oldest challenge is discarded (as storing the whole challenge for freshness is resources consuming, the procedure can be adapted to store only a sub-part of the challenge, e.g., the first 8 bits). Next, the tag calculates k-bits long random value $S \in [0{,}2\ k-1]$. This value serves for rotation for S positions (in shift register) of the least significant $2^k + 1$ bits of the concatenated secret s and the challenge value r. After rotation, the result is hashed to produce a 96 bits long output that is sent to the reader.
3. Upon receipt of the above message the reader starts calculations to find a match. Knowing that the last $k + 1$ positions of the challenge have been randomly rotated, it produces strings $[s\|r_n, r_{n-1}, \ldots, S^0(r_2^k, r_{2-1}^k, \ldots, r_0)] = [s\|r_n, r_{n-1}, \ldots, r_1, r_0]$, $[s\|r_n, r_{n-1}, \ldots, S^1(r_2^k, r_{2-1}^k, \ldots, r_0)]$, $\ldots, [s\|r_n, r_{n-1}, \ldots, S^{2^k-1}(r_k, r_{k-1}, \ldots, r_0)]$ and hashes them ("||" denotes concatenation, s denotes the shared secret, r_i denotes the ith bit in challenge r, and S^a denotes shifting of the last $k + 1$ bits of a challenge for a positions to the left or the right). Afterward it checks the results by looking in a database that contains pairs (ID, s), and when a match is found, the tag is properly identified.

3.5. Required resources for implementation

Analyzing now the new non-deterministic protocol, the following number of NAND gates is obtained:

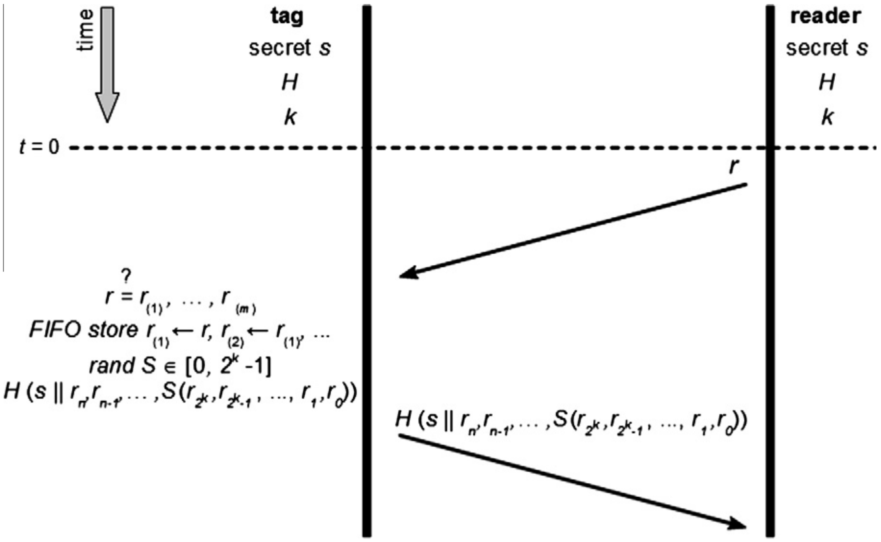
time
tag
secret s
H
k
reader
secret s
H
k
t = 0
r
r =? r(1), ..., r(m)
FIFO store r(1) ← r, r(2) ← r(1), ...
rand S ∈ [0, 2^k -1]
H (s || rn,rn-1, ... ,S(r2k,r2k-1, ..., r1,r0))
H (s || rn,rn-1, ... ,S(r2k,r2k-1, ..., r1,r0))

Fig. 3. Non-deterministic privacy enabling protocol for RFIDs.

- 96-bits are needed for storing tag's identifier (shared secret *s*), which results in 96∗5 = 480 NAND gates.
- Assuming optional prevention of exhaustive challenges attacks and ensuring their freshness, four 96 bit locations for (truncated) challenges are needed in FIFO memory, and this requires 96∗5 = 320 storage cells (to enable as wide range of stored challenges as possible, only the first *k*-bits of each challenge are stored).
- Two shift registers are needed where in the first register the pseudo-random value is generated that is used for shifting of the second register. If the pseudo-random generating shift register has *k* bits, this is sufficient for rotating 2^k bits, so the main cost will be the second register. Therefore for the first register we assume $k = 4$ bits, which means 60 NAND gates, while the main shift register therefore needs $(2^4/4)*60 = 240$ NAND gates, so calculation of the second cryptogram requires total 1260 NAND gates. Alternatively, if a true random generator is used that has been mentioned above, the total number of gates is 1320.
- Hashing can be done by using DESL, which requires approx. 1800 NAND gates (alternatively, if AES is used, approx. 3000 gates are required).

Taking now into account the communication costs (2∗96∗5 = 960 gates), the total approximate number of gates is therefore approx. 2840 + 960 = 3800 gates for the basic option, and approx. 3920 + 960 = 4880 gates for the alternative option with a true random number generator.

With regard to protocols analyzed above, it requires fewer gates than that of Okhubo, Suzuki and Kinoshita, and is not vulnerable to de-synchronization attacks. As to protocol of Henrici and Mueller, it outperforms it as well. The third protocol above (protocol of Weis, Sarma, Rivest and Engels) is also more expensive, while, in addition, the new protocol does not contain its weakness, where the first message is not cryptographically linked to the second one. As to the last protocol above, it has an order of magnitude higher cost than our protocol. Last but not least, the cost of the new protocol is comparable to that of the two non-deterministic protocols presented in [14]. However, if the challenge gets increased, this new non-deterministic protocol starts outperforming the older two, which require five NAND gates for each additional bit of a challenge because of XOR operations (this holds true also for other architecturally similar protocols).

3.6. Security analysis of ND-PEP

Let us now provide an informal security analysis of the protocol:

1. Due to the properties of hash functions (pre-image resistance, second pre-image resistance, collision resistance), the secret that is included in the hashing operation to produce the second message cannot be recovered by an attacker. Therefore as soon as it is exposed to the attacker, the confidentiality of secret *s* is ensured.
2. Authentication is provided on the basis of knowing the secret *s* that is shared with the tag.
3. Authorization of a reader (and the back-end system to which it is connected) is ensured through knowing (having access only to) those pairs (ID, *s*) for which it is authorized.
4. Malicious tracking of a tag is prevented by constantly changing messages in the second step. Further, due to these changing cryptograms (aliases) that form the second message, not only anonymity is provided (non-deterministic values in the second step can be viewed as a kind of aliases), but also reply attacks are prevented. Last but not least, minimizing protocol steps is often prudent practice, because tracking side information of the protocol can reveal the encrypted content [39].
5. The exchanged messages are unique and fresh. This first holds true for the first message, which serves as a parameter for hashing of the second message, where it is concatenated with a secret, and additionally randomly rotated in the last 2^k bit positions. This procedure results in another pseudo-random value, so the both values are always fresh, and they appear random to an attacker. This is aligned with wise practices given in [32].

Summing up, the protocol ensures privacy by prevention of tracking through use of pseudonyms. Further, it prevents excessive triggering of the tag by the reader, ensures reply prevention, and cryptanalytic attacks (in fact, ID never crosses the medium).

Informal analysis is very descriptive, but often not sufficient. Therefore formal analysis of security protocols is often required. In the very particular area of RFIDs an often used formalism is the one proposed by Avoine [40], but we have chosen AVISPA framework [41,42]. The reasons are the following: AVISPA enables tight formalization with its High Level Protocol Specification Language, or HLPSL [43]. Further, it enables automatic verification of HLPSL specifications by using four back-end checkers: On-the-fly Model Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model Checker (SATMC) and Tree Automata-based Protocol Analyzer (TA4SP). Last but not least, AVISPA has obtained a wide reputation in the area of (automated) formal checking and many IETF and ISO standard protocols have been checked by AVISPA. The formalized representation of our protocol, its environment and goals are given in the appendix. We state at the end of this section that ND-PEP has been successfully verified by OFMC (marked as SAFE), CL-AtSe (marked as SAFE), SATMC (marked as SAFE), while TA4SP checker had to give up the verification because of running out of memory.

4. Conclusions

One key challenge in internet of things, IoT, is provision of privacy. Taking into account that IoT devices have numerous limitations, appropriate solutions have to be lightweight, which especially holds true for RFIDs that are in the center of this paper. These devices have the most stringent requirements because of weak computational power and very limited energy (power) resources. On top of this, IoT objects are increasingly penetrating business environments, where the legal pressure is growing when it comes to privacy.

This paper therefore first focuses on the appropriate definition and formalization of privacy that is applicable to business processes by providing a holistic model (based on Petri nets) for formal treatment of privacy in business environments. More precisely, it provides formal means for appropriate addressing of privacy with security policies.

Next, this paper focuses on technological level and presents a new protocol for provision of privacy for computationally weak devices. The harsh technical constraints that have to be met are presented, and followed by details of the new lightweight protocol that is non-deterministic and that deploys computational asymmetry between a tag and a reader. It relies on pseudo-random responses that act as pseudonyms to preserve privacy, and qualifies as being lightweight according to quantitative metrics. Another advantage of this protocol is that it outperforms other architecturally similar protocols. Further, its requirement for additional logic gates grows slower with increased challenge lengths compared to its predecessors.

Acknowledgements

Author acknowledges the support of the Slovenian Research Agency ARRS through program P2-0359 and research project L7-3653 e-Health Support. In addition, this research is a partial result of collaboration in EU project COST WiNeMO.

Appendix A

This appendix gives a formal specification of ND-PEP protocol that was used for formal verification of the presented protocol with AVISPA, which deploys the Dolev–Yao adversary model [44]. The specification is in HLPSL and the details about this language can be found in [42,43] (also without detailed familiarity with HLPSL, its specifications can be mainly understood by those with background in formal verification techniques).

```
role client(
  T,R                    : agent,
  Secret_TR              : symmetric_key,
  H                      : hash_func,
  SND,RCV                : channel(dy))
played_by T def=

local
  State                  : nat,
  Random                 : text

const
  random                 : protocol_id

init
  State                  := 0

transition
1. State                 = 0 ∧ RCV(Random') =|>
State'                   := 1 ∧ SND(H(Secret_TR.Random'))
                         ∧ witness(T,R,random,Random')

end role
```

```
role server (
  T,R                    : agent,
  Secret_TR              : symmetric_key,
  H                      : hash_func,
  SND,RCV                : channel(dy))
played_by R def=

local
  State                  : nat,
  Random                 : text

const
  random                 : protocol_id

init
  State                  := 10
```

```
transition
1. State                          = 10 ∧ RCV(start) =|>
State'                            := 11 ∧ Random':= new()
                                  ∧ SND(Random')

2. State                          = 11 ∧ RCV(H(Secret_TR.Random)) =|>
State'                            := 12
```

```
role session (
  T,R                             : agent,
  Secret_TR                       : symmetric_key,
  H                               : hash_func)
  def=

local
  S1, S2                          : channel (dy),
  R1, R2                          : channel (dy)

composition
  client(T,R,Secret_TR,H,S1,R1)
∧ server(T,R,Secret_TR,H,S2,R2)

end role
```

```
role environment() def=

const
  t,r,i                           : agent,
  h                               : hash_func,
  secret_tr,secret_ir             : symmetric_key

intruder_knowledge                = {t,r,i,secret_ir,h}

composition

session(t,r,secret_tr,h)
∧ session(t,r,secret_tr,h)
∧ session(i,r,secret_ir,h)
∧ session(i,r,secret_ir,h)

end role
```

```
goal

%Reader authenticates Tag on random
authentication_on random

%Confidentiality of shared secret
secrecy_of secret_tr

end goal
```

```
environment()
```

References

[1] Internet of Things Initiative, CASAGRAS Project Description, <http://www.iot-i.eu/iot-database/all/organizations/internet-of-things-initiative/fines-future-internet-enterprise-systems/casagras>.

[2] S. Haller, Internet of things – an integral part of the future internet, SAP Research, <http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf>.

[3] E-practice.eu, Commission launches consultation on radio frequency identification, European Communities, (3.3.2008) <http://www.epractice.eu/document/4426>.

[4] European Commission: Data protection directive, 95/64/EC, Official Journal of the European Communities, L 281, 23/11/1995, Brussels, 1995.

[5] N. Gohring, California makes it a crime to 'skim' RFID tags, PC world, <http://www.pcworld.com/businesscenter/article/151822/california_makes_it_a_crime_to_skim_rfid_tags.html> (accessed 4.05.12), October 3, 2008.

[6] International Standards Organisation, Information technology — security techniques — information security management systems — overview and vocabulary, ISO standard # 27000, Geneva, 2009.

[7] M.C. Tschantz, A. Datta, J.M. Wing, Formalizing and enforcing purpose restrictions of privacy policies, in: Proceedings of 33rd IEEE Symposium on Security and Privacy, May 2012 (forthcomming).

[8] ISO, Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture, ISO standard 7498-2, Geneva, 1989.

[9] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Trans, Comput. Syst. 8 (1) (1990) 18–36 (ACM).

[10] P.F. Syverson, P.C. van Oorschot, On unifying some cryptographic protocol logics, Procedings on the Research in Security and Privacy 94 (1994) 14–28.

[11] G. Bella, S. Bistarelli, Soft constraint programming to analysing security protocols, Theory and Practice of Logic Programming, vol. 4(5-6), Cambridge University Press, 2004. pp. 45–572.

[12] M. Abadi, M. Burrows, B. Lampson, G. Plotkin, A calculus for access control in distributed systems, ACM Transactions on Programming Languages and Systems 15 (4) (1993) 706–734.

[13] D. Trček, P. Japinnen, RFID security, in: Y. Zhang, L. Tianruo, J. Chen, (Eds.), RFID and sensor networks: architectures, protocols, security, and integrations, Taylor & Francis, 2010, pp. 147–168.

[14] F. Finkenzeller, RFID Handbook, John Wiley & Sons, New York, 2010.

[15] NFC Forum, NFC activity specification, NFCForum-TS-Activity-1.0, Edgewater, 2010.

[16] Philips, THE I2C-BUS SPECIFICATION, v2.1, 2000, <http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf> (accessed 4.5.11).

[17] Motorola, SPI Block Guide, v3.06, 2003, <http://www.ee.nmt.edu/≃teare/ee308l/datasheets/S12SPIV3.pdf> (accessed 04.05.12).

[18] A. Juels, RFID security and privacy – a research survey, IEEE Journal on Selected Areas in Communications 24(2) (2006) 381–394, IEEE.

[19] M. Buettner, B. Greenstein, D. Wetherall, Dewdrop: an energy-aware runtime for computational RFID, in: Proc. of the 8th USENIX conference on networked systems design and implementation, NSDI'11, Boston, 2011, pp.15–29.

[20] D.J. Yeager, A.P. Sample, J.R. Smith, WISP: a passively powered UHF RFID tag with sensing and computation, in: S.A. Ahson, M. Ilyas (Eds.), RFID Handbook: Applications, Technology, Security, and Privacy, CRC Press, Boca Raton, 2008.

[21] IEEE, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Spec. for Low Data Rate Wireless Personal Area Networks (WPAN), IEEE Standard 802.15.4, IEEE: Piscataway, 2006.

[22] J.J.P.C. Rodrigues, P.A.C.S. Neves, A survey on IP-based wireless sensor network solutions, Int. J. Commun. Syst. 23 (2010) 963–998.

[23] J. Marinho, E. Monteiro, Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions, Wireless Networks, vol. 18, Springer, 2012, pp. 147–164.

[24] L. Vodovnik, S. Rebersek, Digital circuits, in: Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, 1986.

[25] P. Horowitz, W. Hill, The art of electronics, Cambridge University Press, New York, 1989.

[26] M. Epstein, L. Hars, R. Krasinski, M. Rosner, H. Zheng, Design and implementation of a true random number generator based on digital circuit artifacts, in: Lecture Notes in Computer Science 2779, CHES 2003, Springer, 2003, pp. 152–165.

[27] A. Poschmann, G. Leander, K. Schramm, C. Paar, New light-weight crypto algorithms for RFID, in: Proc. of the IEEE International Symposium on Circuits and Systems – ISCAS 2007, New Orleans, 2007.

[28] B. Schneier, Applied Cryptography, John Wiley & Sons, New York, 1996.

[29] M. Feldhofer, S. Dominikus, J. Wolkerstorfer, Strong authentication for RFID systems using the AES algorithm, in: Lecture notes on computer science 3156, CHES 2004, Springer, 2004, pp. 357-370.

[30] D. Trček, D. Kovač, Formal apparatus for measurement of lightweight protocols, Computer Standards & Interfaces, vol. 09(31), Elsevier, 2009, pp. 305–308.

[31] G. Avoine, RFID security and privacy lounge, <http://www.avoine.net/rfid/> (accessed 16.05.12).

[32] M. Ohkubo, K. Suzuki, S. Kinoshita, A cryptographic approach to a privacy-friendly tags, RFID Privacy Workshop, MIT, November 15, MIT, Cambridge, 2003.

[33] D. Henrici, P. Muller, Hash based enhancement of location privacy for radio frequency identification devices using varying identifiers, in: Proc. of the 1st Int. Workshop on Pervasive Computing and Communication Security, Orlando, 2004, pp. 149-153.

[34] S.A. Weis, S.E. Sarma, R. Rivest, D.W. Engels, Security and privacy aspects of low-cost radio frequency identification systems, in: Proc. of the 1st Security in pervasive computing, LNCS 2802, Boppard, 2004, pp. 201–212.

[35] S.V. Kaya, E. Savas, A. Levi, O. Ercetin, Public-key cryptography based privacy preserving multi-context RFID infrastructure, Ad Hoc Networks, vol. 7(1), Elsevier, 2009, pp. 136–152.

[36] G. Avoine, F. Dysli, P. Oechslin, Reducing the time complexity in RFID systems, in: Proc. of the 12th workshop on selected areas in cryptography, pp.291-306, Kingston, 2005.

[37] G. Avoine, P. Oechslin, RFID traceability: a multilayer problem, in: Proc. of the 9th int. conf. on Financial Cryptography and Data Security, pp. 125-140, Springer 2005.

[38] M. Abadi, R. Needham, Prudent engineering practice for cryptographic protocols, IEEE Transaction on Software Engineering, vol. 22(1), IEEE, 1996, pp. 6–15.

[39] R. Koch, G. Dreo Rodosek, User Identification in Encrypted Network Communications, in: 6th IEEE/IFIP International Conference on Network and Service Management, Niagara Falls, 2010.

[40] G. Avoine, Radio frequency identification: adversary model and attacks on existing protocols, Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology in Lausanne, School of Computer and Communication Sciences, Lausanne, 2005.

[41] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, O. Heám, P.C. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, L. Vigneron, The Avispa Tool for the automated validation of internet security protocols and applications, in: Proc. of CAV 2005, Computer Aided Verification, LNCS 3576, Springer, Berlin.

[42] AVISPA Team, AVISPA User Manual, v 1.1, June 30, 2006, <http://www.avispa-project.org/package/user-manual.pdf>.

[43] AVISPA Team, The high level protocol specification language, deliverable 2.1, <http://www.avispa-project.org/delivs/2.1/d2-1.pdf>.

[44] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on Information Theory IT-29 (2) (1983) 198–208.

Prof. Dr. Denis Trček is with Faculty of Computer and Information Science, University of Ljubljana, where he heads Laboratory of e-media. He has been involved in the field of computer networks and IS security and privacy for over 20 years. He has taken part in various EU and national projects in government, banking and insurance sectors (projects under his supervision totaled to approx. one million EUR). His bibliography includes over one hundred titles, including monograph published by renowned publisher Springer. D. Trček has served (and still serves) as a member of various international bodies and boards (MB of the European Network and Information Security Agency, etc.). His interests include security, trust management, privacy and human factor modeling. He is a member of IEEE and can be reached at denis.trcek@fri.uni-lj.si.