Privacy Preserving Products Tracking in Clustered Supply Chain

Mehdi Khalfaoui, Kaoutar Elkhiyaoui, Refik Molva
Institut Eurecom
2229, Route des Cretes
06560 Valbonne, France
Email: mehdi.khalfaoui@eurecom.fr, kaoutar.elkhiyaoui@eurecom.fr, refik.molva@eurecom.fr

***Abstract*—One of the main applications of supply chain management is product tracking. We define it as tracing the product path along the supply chain. In this paper, we propose a solution to track the product while preserving the privacy of the supply chain actors involved and the path traced. More precisely, this solution allows to identify which path a product has taken in the supply chain, without disclosing sensitive information. To allow product tracking, the product are attached to a sensor node. This latter stores a trace of the product path along the supply chain. The trace is computed using polynomial based signature techniques. We restrict the visibility of the manager of the supply chain by organizing the supply chain facilities into clusters. Also, we encrypt the path traces to ensure security against adversaries. To perform access control in the sensor nodes we use randomized Rabin scheme which is known for being efficient and lightweight. In this paper, sensor nodes are not required to perform heavy computation, which makes our solution feasible. The main achievement of this work is a cryptographic mechanism that allows to the supply chain manager to trace the supply chain entities that product went through, without disclosing the identity of those entities.**

***Keywords*-Supply Chain; Privacy; Tracking; Cluster.**

I. Introduction

Recently, sensor-based applications have become more and more popular. One of their major applications is supply chain management [21]. More precisely, sensor nodes are used to monitor and track products from production, storage to distribution [6]. Furthermore, the heterogeneity of different parties involved in the supply chain, raises new security and privacy challenges. Partners aim at verifying the legitimacy of products in their sites, yet they are reluctant into leaking information about their internal processes.

Studies show that most security incidents involve business partners. This a significantly growing trend over the last few years [2]. Furthermore, the number of security incidents affects the trust between partners in supply chains [2]. Therefore, there are many attempts to overcome security issues and the lack of trust among the partners in the supply chain. In [14], the authors propose a mechanism based on Secure multi-party computation [26]. They aim at achieving security against the business partner. It enables the supply chain partners the computation of joint function without disclosing their inputs data. Only the final input is revealed. Secure multi-party computation can only compute specific families of functions, which restricts the used operations. In our case, Secure multi-party computation cannot compute our path trace. Elkhiyaoui et al. [4] propose RFID-based tracking mechanisms of the products in the supply chain. They allowed to authorized entities to validate the path of the products, without disclosing the identity of the others partners in the supply chain. The issue with this approach is the use of unsuitable arithmetic operations for low capacity devices.

In this context, the paper at hand aims at enabling the manager of the supply chain to verify the validity of the path a product took. Such a verification could allow the manager to detect counterfeits in the supply chain. Here, we assume that each supply chain has its own global manager. However, supply chains are distributed over sites or facilities. The latter reside in different locations and belong to different partners. Hence, the supply chain manager does not have full control over interconnections among the facilities. He also does not have full control over some of the facilities themselves. We propose, as possibility, that the products carry necessary information that will allow to a manager to verify their paths. To this effect, we use the memory capacity of the sensor nodes that are attached to the products to store the path trace.

In this paper, we propose a mechanism to protect the partners' privacy and the product privacy in the supply chain. First, to protect partners privacy against the manager, this latter should have a restricted visibility of the partners internal processes. Thus, we organize the supply chain into clusters such where sites and facilities belonging to the same partner will form a single entity. Then, when product arrives to the manager, he can identify which partners handled it. However, he cannot know precisely which sites and facilities the product visited. Figure 1 illustrates a clustered supply chain. This supply chain consists of : production, storage, and distribution cluster. Each cluster consists of three sites.

Second, to protect product privacy in the supply chain, only the manager should be able to verify the path of products in the supply chain.

The research was partially funded by the German Federal Ministry of Education and Research under the promotional reference 01ISO7009 and by the French Ministry of Research within the RESCUE-IT project. The authors take the responsibility for the content

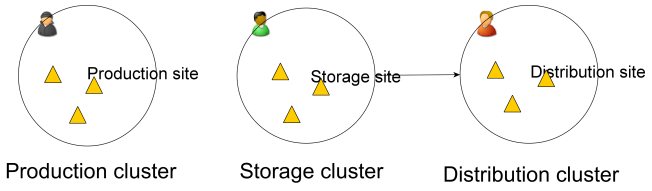

Figure 1. clustered supply chain

Any technical solution that addresses secure and privacy preserving product tracking should take into account the limitations of sensor nodes. These are constrained devices in terms of computation, power and memory. For example, RSA signature [8] cannot be implemented in sensor nodes.

This paper introduces a mechanism to track products in supply chain while protecting sensitive information of supply chain partners and products. The main idea is to organize the supply chain into clusters to restrict the manager visibility, and to attach products to sensor nodes which store an encoding of the products' path. The path encoding is computed using polynomial based signatures for run time fault detection [17]. In order to ensure the validity of the actors that interact with the product, sensors use Rabin scheme to authenticate them. However, Rabin scheme can be easily replaced by other mechanisms such as the one proposed by Gomez et al. [24]. Then, The path trace is encrypted to ensure confidentiality.

The major contributions of this work:

- It allows the supply chain manager to verify the validity of the paths a product took. More precisely, it allows the manager to verify which sequence of clusters, a product have visited.
- It guarantees the privacy of products and therewith partners in the supply chain. Only the manager is able to verify the path the product took.
- It only requires sensors to perform efficient low capacity operations and to store few Kbytes.
- It allows the restriction of product information that can be disclosed to each partner.

II. Solution

A. Preliminaries

A supply chain in this paper simply denotes a set of sites that a product goes through. As discussed previously, we organize the supply chain into clusters such that sites belonging to the same partner belong to the same cluster. These clusters are used to restrict the visibility of the manager of the supply chain. The manager is able to recognize the cluster a product visited but not the exact site the product visited. Thus, in this manner we enforce the privacy of the partners in the supply chain against the manager of the supply chain. The solution proposed involves the following entities:

- **Sensors** S_i: each sensor is attached to a product in the supply chain. A sensor S_i is equipped with a rewritable memory that stores the trace $s_{(S_i,j)}$. $s_{(S_i,j)}$ represents the trace of the path that the sensor took in the supply chain until cluster j. Sensors can also compute a cryptographic function f to authenticate the partners' systems in the supply chain.
- **Manager** M: the manager M attaches S_i to a product and writes into S_i an initial state $s_{(S_i,0)}$. M wants to identify the path a sensor S_i went through. More precisely, M wants to identify the sequence of clusters that S_i went through. M therefore, reads the current state $s_{(S_i,j)}$ of S_i, and decides whether S_i visited legitimate sequence of clusters or not. We assume that M knows which paths in the supply chain are valid or not. In other words, M has a database DB of valid path traces.
- **Clusters** c_k: To enforce the privacy of the partners in the supply chain, the supply chain will be organized into clusters. Each cluster contains a set of supply chain sites that belong to the same supply chain partner. Without loss of generality, we assume that each cluster c_k is equipped with a supply chain actor's system A_k. A_k uses some function f_{A_k} to generate $s_{(S_i,j+1)}$ from $s_{(S_i,j)}$, i.e., $f_{A_k}(s_{(S_i,j)}) = s_{(S_i,j+1)}$. The actor's system A_k can also compute a cryptographic function g to authenticate themselves to sensors.

III. Protocol

A. Protocol overview

In this paper, a sensor S state noted $s_{(S,j)}$ represents the sequence of clusters in the supply chain that S visited. One of the challenges in this work is to encode the sensor's path efficiently, i.e. encoding ha sto be independent of the number of visited clusters by the sensor. For that purpose, we use a technique for run time fault detection to encode paths using polynomials. More precisely, each valid path in the supply chain P_{valid} will match the evaluation of a unique polynomial $Q_{P_{\text{valid}}}(x_0)$ at a fixed number x_0. The efficiency of this encoding relies on two properties: **1)** a path is represented as polynomial evaluation at point x_0, therefore, the size of the encoding does not depend on the number of clusters a sensor visited **2)** for any two different paths in P_1 and P_2, the equation $Q_{P_1}(x_0) = Q_{P_2}(x_0)$ holds only with negligible probability [17]. As a result, the state of a sensor node S at the end of the supply chain can be uniquely mapped to one single path.

However, the path representation as presented above does not suffice to prevent path cloning, i.e., copying the path of a valid sensor into a fake sensor and then injecting the fake sensor in the supply chain. To tackle this problem, sensors store $Q_{P_{\text{valid}}}(x_0)$ added to a keyed HMAC of their unique IDs. HMAC is used for two purposes: first, it ensures that sensors are issued by a legitimate authority and prevents an

adversary from injecting its own sensor. Second, it allows to map the Sensor's ID to a random number that cannot be predicted by the adversary. Therefore, an adversary cannot clone a sensor more than once, and thus, cloning cannot be performed in a large scale.

Whenever, a sensor S visits a cluster in the supply chain, the actor's system updates the sensor node's state by updating the polynomial evaluation. In a nutshell, the protocol consists of

- Initialization phase: Supply Chain manager M initializes sensors, and distributes IDs to the different partners.
- Authentication phase: Sensor S authenticates each visited site, before updating its trace.
- Collection phase: Sensors successively store the evaluation of a polynomial. That is achieved by updating the trace of the sensor in each cluster.
- Verification phase: Manager M extracts the sensor's path trace and therewith the polynomial. M checks whether the sensor state corresponds to a valid sequence of clusters.

Privacy and security overview: : To protect product privacy, sensors will store only probabilistic Paillier encryptions [19] of their states, and the actor's systems use homomorphic techniques for arithmetic operations on encrypted path encodings. At the end of the supply chain, M can then decrypt and identify the path. Also, the use of Paillier cryptosystem and HMAC guarantees the security of path encoding.

B. Path Encoding technique

The polynomial path encoding is used in [4]. It is based on techniques for software fault detection. Noubir et al. [17] propose to encode a softwares state machine using polynomials such that the exact sequence of states visited during run-time generates a unique *"mark"*. Therewith, run-time faults can be detected. By considering the actor's system instead of state machine, the path encoding used by Noubir et al. [17] can be applied in our case.

For each cluster c_i in the supply chain, c_i is associated with a unique random identifier $c_i \in \mathbb{F}_q$, where q is a large prime.

As mentioned above, a path in the supply chain is represented as a polynomial $\in \mathbb{F}_q$. The polynomial corresponding to a path $\mathcal{P} = \overrightarrow{c_0 c_1 \dots c_l}$ is defined in Equation (1). All operations are in $\mathbb{F}_q$.

$$Q_{\mathcal{P}}(x) = \sum_{i=0}^{l} c_i x^{l-i} \tag{1}$$

To have a more compact representation of paths, a path $\mathcal{P}$ is represented as the evaluation of $Q_{\mathcal{P}}$ at x_0, where x_0 is a generator of $\mathbb{F}_q^*$. We denote $\phi(\mathcal{P}) = Q_{\mathcal{P}}(x_0)$. The desired property of anti-collision, .i.e $\forall \mathcal{P} \neq \mathcal{P}\prime, Pr(\phi(\mathcal{P}) = \phi(\mathcal{P}\prime)) = \frac{1}{q}$ [17], ensures the uniqueness of the path mark with high probability.

C. Paillier Cryptosystem

The following is description to the Paillier cryptosystem [19] that we use in order to achieve both privacy and security of our mechanism:

Key Generation: Let k be the security parameter. Choose uniformly and at random two k-bit primes p and q, set $N = pq$, and set $\lambda(N) = lcm(p-1, q-1)$. Choose a random base $g \in \mathbb{Z}_N^*$.

Encryption: To encrypt message $m \in \mathbb{Z}_N$, one chooses a random value $r \in \mathcal{Z}_N^*$ and computes the ciphertext as

$$c = \mathcal{E}(m, r) = g^m r^N mod N^2 \tag{2}$$

Decryption: When receiving a ciphertext c, check that $c < N^2$. If yes, retrieve the message m as

$$m = \mathcal{D}(c) = \frac{L(c^{\lambda(N)} mod N^2)}{L(g^{\lambda(N)} mod N^2)} mod N \tag{3}$$

Where $\forall u \in \{u < N^2 / u \equiv 1 mod N\} L(u) = \frac{u-1}{N}$

Additive Homomorphic property: Paillier cryptosystem has the property to be additively homomorphic:

$$\mathcal{E}(m_1, r_1) * \mathcal{E}(m_2, r_2) = \mathcal{E}(m_1 + m_2, r_1 r_2) \tag{4}$$

This property allows the execution of arithmetic operations on encrypted data. Therefore, it supports the evaluation the polynomial mark at each cluster of the supply chain without decryption.

Self Blinding: Paillier cryptosystem has the property to be *Self-Blinding*, i.e the property by which any ciphertext can randomly be changed into another without affecting the plaintext. This property is achieved as follows:

$$\forall r \in \mathbb{Z}_N \mathcal{D}(\mathcal{E}(m, r)) = m \tag{5}$$

Therefore, the decryption of any message m is independent of the value of r.

D. Detailed Protocol description

Our protocol consists of an initialization phase, the phase that prepares a sensor to enter the supply chain. Then, the authentication phase to verify the legitimacy of the site before collecting its trace. Collection phase, where the sensor interacts with different Sites and collects its traces. Finally, the path verification phase, when the polynomial mark is extracted by the supply chain manager M and the path gets verified. Figure 2 illustrates the sequence of the different phases, which compose the sensor life-cycle.

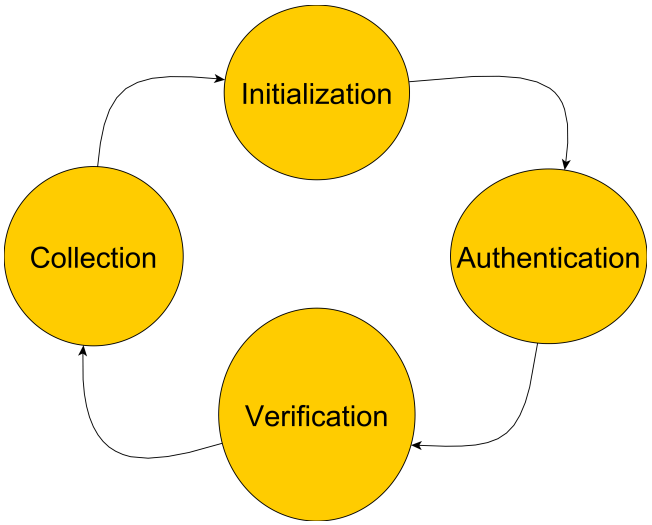

Figure 2. sensor lifecycle

1) Initialization phase: In this phase, we assume that every partner in the supply chain has the necessary resources to perform the following actions:

- supply chain manager M shares with the supply chain actors a Rabin's private key (p_R, q_R). Rabin cryptosystem [22] is used to achieve authentication of the supply chain actors. Rabin encryption is single square modular encryption, which makes it feasible for low capacity devices such as sensor nodes. Rabin's public key $N_R = p_R q_R$ is stored in the sensor to perform the authentication process.
- M generates randomly a list of cluster identifiersc_{list}. For the sake of simplicity, we don't distinguish between cluster c_i and its identifier.
- M generates a Paillier public key pk_M and private key sk_M. Then M sends $\mathcal{E}_M(c_i)$ to each actor belongs to the cluster c_i in a secure way.
- M generates the identifiers list S_{list} of sensors. For the sake of simplicity, we assimilate a sensor S_i and its identifier ID_i. M stores in each sensor S_i, the Paillier encryption of its ID_i, the Paillier encryption of $HMAC_k(S_i)$, where $HMAC_k$ is keyed hash function [3], and k is its secret key.

Now, for each sensor entering the supply chain, M has already stored on it the initial value $s_{(S_i 0)} = E(HMAC_k(S_i))$. The table I illustrates the exchanges messages in the initialization phase between M and S in , and between M and each site in the supply chain.

2) Authentication phase: In this phase, the sensor S and the actor's system A interact in the supply chain by executing the following actions: We assume that the sensor S has visited the clusters c_0, ..., c_l. When, S visits the actor's system A_{l+1}, it is already stored the trace $\mathcal{P}_l = \overrightarrow{c_1 c_2 ... c_l}$ that encodes the path from the sites that belongs to the clusters

$M \rightarrow S$, $g^{HMAC_k(S)}$
$\mathcal{M} \rightarrow Site \in c_i$, $\mathcal{E}_M(c_i)$

Table I
INITIALIZATION PHASE

$c_1, c_2, \ldots, c_l$. Therefore, the current state of the sensor is $s_{(S,l)}$, which corresponds to the state of the sensor after interacting with l clusters .

S chooses a random value $r \in \mathbb{F}_{N_R}$ and sends $Rabin(r) = r^2 mod N_R$ to A, while storing the $hash(r)$ and $hash(N_R - r)$. Abduvaliev et al. [1] show that a large possibility of hash functions can be implemented in low capacity devices such as sensor nodes. A decrypts $Rabin(r)$ using its public key. The decryption gives exactly four solutions, r, $N_R - r$, t, $N_R - t$. As the actor does not know which is the real solution, he chooses to send back to S two hash values. The values are chosen in such a way that their sum is not null $mod N_R$. For example A chooses to send $hash(r)$, and $hash(N_R - t)$. Therefore, S considers the authentication as successful, if one of the received value matched one of the stored value. Then, S can start trace collection procedure. Table II illustrates the messages exchanged between S and site to verify if the actor is legitimate or not.

$\mathcal{S}$ picks randomly a number r
$\mathcal{S} \rightarrow Site$, $r^2 mod N_R$
$Site \rightarrow S$, $hash(r), hash(t)$

Table II
AUTHENTICATION PHASE

3) Collection phase: After the authentication phase, S starts the collection phase. S sends its current state $s(S,l)$ to the actor's system A_{l+1}. A_{l+1} updates the sensor's state as following:

$$s(S,l+1) = s(S,l)^{x_0} * \mathcal{E}_M(c_{l+1}) = s(S,l)^{x_0} * g^{c_{l+1}} r^N mod N^2 \quad (6)$$

Assuming that our products has to interact with n supply chain partner, the final sensor's state is:

$$\begin{aligned} s(S,n) &= s(S,n-1)^{x_0} * g^{c_n} r_1^N mod N^2, where \quad r_1 \in \mathbb{F}_N \\ &= s(S,0) * g^{\sum_{i=1}^{n} c_i {x_0}^{n-i}} r_2^N mod N^2, where \quad r_2 \in \mathbb{F}_N \\ &= g^{HMAC_k(S) x_0^n + \sum_{i=1}^{n} c_i {x_0}^{n-i}} r_3^N mod N^2, \quad r_3 \in \mathbb{F}_N \\ &= \mathcal{E}_M(HMAC_k(S) {x_0}^n + \sum_{i=1}^{n} c_i {x_0}^{n-i}) \end{aligned}$$

Table III illustrates the two messages exchanges between S and random site in the collection phase.

$S \rightarrow$ Site, $s(S,i)$
Site $\rightarrow S$, $s(S,i+1)$

Table III
COLLECTION PHASE

4) Verification phase: : In this phase, the supply chain manager M checks if the path recorded in the sensor is a valid one. M extracts the final state from the sensor $s(S,n)$, and decrypts it, so he can extract the path trace $\phi(\mathcal{P})$.

$$\begin{aligned} \phi(\mathcal{P}) &= \mathcal{D}_{TTP}(s(S,n)) = \sum_{i=1}^{n} a_i x^i \\ &= HMAC(S)x_0^n + \sum_{i=1}^{n} c_i {x_0}^{n-i} \end{aligned} \tag{8}$$

Using successive division operations, M extracts the coefficients a_0,a_1,...,a_n of the polynomial $\phi(\mathcal{P})$. Then, M computes $HMAC_k(S)$ and compare it with a_n. If $a_n = HMAC_k(S)$, M accepts the sensor. Otherwise, M rejects the sensor. Finally, M checks if the cluster identifiers c_1, c_2, ..., c_n belongs to the cluster identifiers list, and the path trace $\phi(\mathcal{P})$ proofs that the sequence of the clusters is valid. if one of the identifiers, or the sequence is not valid, M rejects the sensor, and declares the products as non compliant.

IV. EVALUATION

Our protocol is implementable using today's sensors such as Crossbow motes [13] and phidgets [12]. It only requires sensors to store the Rabin public key, which is 1024 bits, and the encrypted state, which is 2048 bits, so a total memory of 3Kb. Through the different steps of the supply chain, the amount of memory needed does not increase. Storing 3Kb of data is feasible in today's sensor hardware. Hempstead et al. [10] show that the memory available in hardware sensors are between 8 kB (i.e. 68Kb) and 132 kB (i.e. 1056 Kb). Therefore, from a memory capacity perspective, our protocol is efficient.

The complexity on the nodes is low. The sensor has to perform at each step of the supply chain, the same arithmetic operations. Therefore, the complexity is linear to the number of the actors in the supply chain. The sensor needs to authenticate the visited site in the supply chain,which means it has to perform one modular square and to compute two hash functions. These operations are necessary to perform a Rabin based authentication. Gaubatz et al. [7] showsthat one (7) modular multiplication in sensor, needs roughly $1\mu J$, which is very low compared to RSA signature requirements [25]. A sensor node of type crossbow has 2 AA batteries, which means roughly $4KJ$ of energy [11].

V. RELATED WORK

The idea of using WSN in the supply chain management to track goods was first suggested in [16]. However, research focused mainly on RFID tags to achieve secure tracking in supply chain. Ouafi and Vaudenay [18] address counterfeiting of products using strong cryptography on RFID tags. Blass et al. [4] presente a tracker, a new mechanism to protect against malicious state update of tags in each step of the supply chain. Secure tracking of specific target using WSN was also addressed in [9]. It describes a mechanism of tracking a moving target based on relaxation algorithms [20]. However, passive RFID tags have limited ressources, which makes security and privacy hard to achieve. As matter of fact, Modular multiplication which is necessity to perform arithmitic operations, cannot be impleted in this type of RFID tags. Only hash functions is implementable in passive RFID tag environment. Chawla et al. [5] check whether covert channels exist in a supply chain that leak information about a supply chains internal details to an adversary using security mechanism implemented in RFID tag. Therefore, tags state is frequently synchronized with a backend-database. If a tags state contains data that is not in the database, the tag is rejected. As supply chain involve different actors, it is difficult to have a single backend-database common among them. Our mechanism focus, however, is on secure and privacy-preserving identification of the path a sensor has taken. Shuihua and Chu [23] detect malicious tampering of a tags state in a supply chain using watermarks. However, there is neither a way to identify a tags path, nor to protect its privacy in the supply chain. Kerschbaum and Oertel [15] detect counterfeits in the supply chain using pattern matching for anomaly detection. This latter can be combined with our mechanism to achieve cloning countermeasure.

VI. CONCLUSION AND OUTLINE

In this paper, we presented a protocol to secure the tracking of products in supply chain. Our main idea is to encode the path of the products using polynomial path encoding. Partners in the supply chain update the path trace successively, such that the path has unique identifier. Our protocol's security and privacy proprieties relies on the semantic security of Paillier and the security of HMAC. It requires only one modular multiplication in each step, and only 3Kb of storage, which ensures its feasibility in available sensors in the market.

In our supply chain scenario, we assume that we have a global supply chain manager. There is no notion of multiple managers. However in real world, That is might not be true.

Supply chain can have a quality, security, and recall manger. Delivering the right information to the right manager is an issue, especially in big scale supply chains. However, this is left to future work

REFERENCES

[1] A. Abduvaliev, S. Lee, and Y. Lee. Simple hash based message authentication scheme for wireless sensor networks. In *Proceedings of the 9th international conference on Communications and information technologies*, ISCIT'09, pages 982–986, Piscataway, NJ, USA, 2009. IEEE Press.

[2] W.H. Baker, C.D. Hylender, and J.A. Valentine. Data breach investigations report. *Verizon Business RISK Team*, 2008.

[3] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in CryptologyCRYPTO96*, pages 1–15. Springer, 1996.

[4] E. Blass, K. Elkhiyaoui, and R. Molva. Tracker : security and privacy for rfid-based supply chains. In *NDSS'11, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011, San Diego, California, USA, ISBN 1-891562-32-0*, 02 2011.

[5] K. Chawla, G. Robins, and W. Weimer. On Mitigating Covert Channels in RFID-Enabled Supply Chains. *RFIDSec Asia, Singapore*, 2010.

[6] A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, pages 475–490, 1960.

[7] G. Gaubatz, J.P. Kaps, and B. Sunar. Public key cryptography in sensor networksrevisited. *Security in Ad-hoc and Sensor Networks*, pages 2–18, 2005.

[8] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-based undeniable signatures. *Advances in CryptologyCRYPTO'97*, pages 132–149, 1997.

[9] R. Gupta and S.R. Das. Tracking moving targets in a smart sensor network. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, pages 3035–3039. IEEE, 2003.

[10] M. Hempstead, M.J. Lyons, D. Brooks, and G.Y. Wei. Survey of hardware systems for wireless sensor networks. *Journal of Low Power Electronics*, pages 11–20, 2008.

[11] http://www.allaboutbatteries.com/Energy tables.html. Last access: 05/06/2011.

[12] http://www.phidgets.com/. Last access: 12/06/2011.

[13] http://www.xbow.com/. Last access: 01/06/2011.

[14] F. Kerschbaum and R.J. Deitos. Security against the business partner. In *Proceedings of the 2008 ACM workshop on Secure web services*, pages 1–10. ACM, 2008.

[15] F. Kerschbaum and N. Oertel. Privacy-preserving pattern matching for anomaly detection in RFID anti-counterfeiting. *Radio Frequency Identification: Security and Privacy Issues*, pages 124–137, 2010.

[16] W. Liu, Y. Zhang, W. Lou, and Y. Fang. Managing wireless sensor networks with supply chain strategy. 2004.

[17] G. Noubir, K. Vijayananda, and H.J. Nussbaumer. Signature-based method for run-time fault detection in communication protocols . *Computer Communications*, pages 405–421, 1998.

[18] K. Ouafi and S. Vaudenay. Pathchecker: An RFID Application for Tracing Products in Supply-Chains. In *International Conference on RFID Security*. Citeseer, 2009.

[19] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in CryptologyEUROCRYPT99*, pages 223–238. Springer, 1999.

[20] K.R. Pattipati, S. Deb, Y. Bar-Shalom, and R.B. Washburn Jr. A new relaxation algorithm and passive sensor data association. *Automatic Control, IEEE Transactions on*, pages 198–213, 1992.

[21] EU Project. Stop tampering of products. 2010.

[22] M.O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. 1979.

[23] H. ShuiHua and C.H. Chu. Tamper Detection in RFID-Enabled Supply Chains Using Fragile Watermarking. In *RFID, 2008 IEEE International Conference on*, pages 111–117. IEEE.

[24] A. Sorniotti, R. Molva, and L. Gomez. Efficient access control for wireless sensor data. *Ad Hoc & Sensor Wireless Networks*, pages 325–336, 2009.

[25] L. Uhsadel, A. Poschmann, and C. Paar. Enabling full-size public-key algorithms on 8-bit sensor nodes. In *Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks*, pages 73–86. Springer-Verlag, 2007.

[26] A.C. Yao. Protocols for secure computations. In *proceedings of the 23rd Annual IEEE symposium on Foundations of Computer Science*, pages 160–164. Citeseer, 1982.