On Concealed Data Aggregation for WSNs

Steffen Peter and Krzysztof Piotrowski and Peter Langendoerfer
IHP GmbH, Frankfurt(Oder), Germany
Email: {peter,piotrowski,langendoerfer}@ihp-microelectronics.com

***Abstract*— In this paper we discuss algorithms that allow the concealed data aggregation (CDA) in wireless sensor networks. We describe and evaluate three algorithms that were reported to suit to the WSN scenario. As result of the evaluation, where we emphasize the awareness to potential attack scenarios, we present a brief overview of strengths and weaknesses of the algorithms. Since no algorithm provides all desirable goals, we propose two approaches to cope with the problems. The first is the successive combination of two algorithms. It increases security, while the additional efforts can be minimized by carefully selected parameters. For the second approach we face specific weaknesses and engineer mechanisms that solve the particular issues. With the considered homomorphic message authentication code and a discussion of the id-issue we exemplary evaluate the two biggest issues of the very promising CMT algorithm.**

I. Motivation

Reducing the total required energy in a wireless sensor network is an outstanding goal. Beside the power required for the computation on the nodes, the power needed for sending and receiving the data packets in the network is a significant factor. Sending one bit requires the same amount of energy as executing 50 to 150 instructions on sensor nodes [11]. Thus, omitting as much network traffic as possible is a substantial task in the area of designing WSN applications.

A well known approach, which is the basis for the following investigations, is the in-network aggregation (INA). In a WSN sensed values should be transmitted to a sink. In many scenarios the sink does not need the exact values for all sensors but a derivative such as sum, average, or deviation. The idea of the INA is to aggregate the data required for the determination of the derivatives as close to the source as possible instead of transmitting all sensed values through the entire network.

A serious issue connected with the INA is the security of the data. Considered that the data is transmitted encrypted, there is the problem that all aggregation nodes, i.e. the sensor nodes that perform the actual aggregation in the network, must have access to the decrypted values. Beside the lack of end-to-end (ETE) security, such a hop-by-hop (HBH) encryption as it is for example part of TinySec [5] has the drawback that the data must be decrypted and re-encrypted on every aggregation node. An approach that promises the combination of ETE-security and INA is the concealed data aggregation (CDA).

The work described in this paper is based on results of IST FP6 STREP UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

II. Background

CDA is an improved version of the INA, which in contrast to the classic HBH ensure the ETE-privacy, i.e. the encrypted values do not need to be decrypted for the aggregation. Instead, the aggregation is performed with encrypted values and only the sink can decrypt the result. Indeed, such an approach requires sophisticated cryptographic algorithms and properties, we will dwell on later.

Considered that such a secure INA exists, it has significant benefits compared to HBH and classic ETE encryption.

1) Network traffic: One major benefit of CDA is its efficiency of both computation effort and network traffic. Since the data is aggregated in the network, the network efficiency is better than ETE without aggregation. In [2] network configurations are described that reduce the network traffic by 85% due to CDA. In order to improve the network efficiency the packet size must be considered. Large encrypted packets could negate the positive network effect.

2) Computation effort: Compared to the HBH-aggregation, the computation effort can be assumed as smaller, because there is no need for decryption and encryption on the aggregation nodes. Indeed, this is only true if the cryptographic algorithms that allow the concealed aggregation do not require too many additional computations.

3) Security: Another benefit is the improved security in comparison to the HBH-aggregation. Since the values are not decrypted on every aggregation node, there are less points where an adversary could catch the unencrypted values.

The fundamental basis for CDA are cryptographic methods that provide the privacy homomorphism (PH) property. An encryption algorithm E() is homomorphic, if for given E(x) and E(y) one can obtain E(x$*$y) without decrypting x,y for some operation $*$. The concept was introduced by Rivest et. al [12] in 1978. The two most common variations of PHs are the additive PH and the multiplicative PH. The latter provides the property $E(x \times y) = E(x) \otimes E(y)$. Well known examples of multiplicative PHs are RSA and the discrete logarithm ElGamal. But since the multiplicative aggregation does not have apparent applications in the field of INA on WSNs, we restrict our search to an efficient PH to the additive PHs with the property $E(x + y) = E(x) \oplus E(y)$.

III. State of the Art

Several PH algorithms have been proposed in the literature. A large subgroup is the family of high degree residuosity class based cryptographic algorithms, for example from Paillier [9], Naccache-Stern[7], and Okamoto-Uchiyama [8]. Even though these public key schemes provide the additive PH, we do not analyze them in this work, because they need very long keys that imply large messages and computation effort that does not suit the WSN scenario. Embodiments of these schemes, designed as public-key elliptic curve discrete-log encryption scheme, were introduced in [10]. But according to [6] this interesting approach still requires too much bandwidth and computation efforts.

Below we discuss three approaches that fit to the WSN world.

A. Domingo-Ferrer (DFPH)

In [3] Domingo-Ferrer introduced a symmetric PH (DFPH) scheme that also has been proposed as efficient PH cryptographic system for WSNs in [4].

Domingo-Ferrer (2002) Algorithm [3]	
Parameter:	*public key: integer* $d \geq 2$*, large integer* M *secret key:* g *that divides* M*;* r *so that* r^{-1} *exists in* $\mathbb{Z}_M$
Encryption:	split m into d parts $m_1..m_d$ that $\sum_{i=1}^{d}(m_i) \bmod g = m$ $C = [c_1, ..., c_d] = [m_1 r \bmod M, m_2 r^2 \bmod M, .., m_d r^d \bmod M]$
Decryption:	$m = (c_1 r^{-1} + c_2 r^{-2} + ... + c_d r^{-d}) \bmod g$
Aggregation:	Scalar addition modulo M $C12 = C1 + C2 = [(c1_1 + c2_1) \bmod M, ..., (c1_d + c2_d) \bmod M]$

It has both the additive and the multiplicative PH property. It is a symmetric algorithm that requires the same secret key for encryption and decryption. The aggregation is performed with a key that can be publicly known. It is required that the same secret key is applied on every node in the network. The message size is $d \cdot n$ bit, so that for very secure parameter combinations (d>100) the messages become very big [13].

B. CMT - a Key stream based PH

A key stream based PH was proposed in [2] by Castelluccia, Mykletun, and Tsudik. We denote it CMT, corresponding to the authors initials. It applies individual keys on every node and promises provable security with small ciphertext sizes.
The idea is to perform a modular addition of a classic stream cipher and with the sensed data. Every sensor uses a different pseudo random stream, for example RC4 or AES in CBC modus. For encryption, the plaintext is simply added to the current key of the stream modulo the length of the key space M. The sink has to subtract the corresponding key stream to obtain the plaintext again.

Since the message size is determined by M, and only one modular addition is required for encryption and aggregation, CMT is very well suited for the application on WSNs. A problem is the decryption, which requires exactly the same key stream as at each sensor node. It is not only a potential computation problem, but the sink that decrypts the aggregated values must also know which sensor data is part of the aggregate. It has to subtract exactly the same key streams that have been used for the aggregation. The knowledge of these nodes is substantial for the algorithm. This ID-problem we discuss later in detail.

Castelluccia, Mykletun, Tsudik (CMT) Algorithm [2]	
Parameter:	*select large integer* M
Encryption:	*Message* $m \in [0, M-1]$, *randomly generated keystream* $k \in [0, M-1]$ $c = (m + k) \bmod M$
Decryption:	$m = (c - k) \bmod M$
Aggregation:	$c_{12} = (c_1 + c_2) \bmod M$

C. Elliptic Curve ElGamal

In contrast to the both PHs presented so far, the elliptic curve ElGamal (ECEG) based PH is an asymmetric cryptographic approach. The benefit of this PH is that the encryption key may be publicly known. As the name suggests the ECEG PH is based on the well investigated ECEG cryptographic algorithm.

ECEG PH Algorithm [6]	
Parameter:	*private key integer* x *public key (G,H), G and H are points on EC, H=xG*
Encryption:	$C = [c_1, c_2] = [kG, kH + mG]$ = *tuple of EC points*
Decryption:	$mG = (kH + mG) - x(kG)$ *demap:* $mG \rightarrow m$
Aggregation:	scalar EC-point addition $C12 = C1 + C2 = [(c1_1 + c2_1), (c1_2 + c2_2)]$

ECEG introduces a serious issue. The message text must be mapped on the EC. In [1] and [6] an approach has been proposed that multiplies the message text with the generator of the EC. It is also our preferred mapping algorithm, even though it causes some problems. The decryption leads again to the mapped point mG, but it is not trivial to compute m out of mG. Since it is the fundamental property of ECC that the point multiplication is not efficiently invertible, the only solution is a brute force computation that relies on a limited domain of the mapping. In most cases this approach is very reasonable. Please noticed that without a valid key it is not even feasible to compute the point mG, so that the security is not interfered by the de/mapping.

IV. Evaluation

In this section we evaluate the described approaches regarding their resistance against diverse attack scenarios. The evaluation is followed by a short overview of the properties.

A. Attack scenarios

The most important property of a cryptographic scheme is its security. Security means resistance against attacks. In the following and address the major attack scenarios for CDA schemes for WSNs and evaluate to what extent the considered cryptographic schemes have the desired resilience.

1) Passive attacks: These are the most important attacks. The adversary does nothing but listening to the transmitted packets. The primary security goal is that such an adversary is not able to gain any information by simple eavesdropping.

a) Ciphertext analysis: A very common attack is the analysis of encrypted packets. A secure cryptographic system must ensure that an adversary is not able to decide whether a encrypted packet corresponds to a specific plaintext or not. Applied properly, the three described CDA schemes are secure against this kind of attack. If the secret keys are hidden, there is no way to obtain information out of the encrypted data.

b) Known plaintext attack: In this kind of attack the adversary tries to determine the secret information with the additional knowledge of the plaintext. In a WSN scenario such an attack is likely since an attacker can obtain plaintext, e.g. by own sensor, physically accessing the deployed sensor, or manipulating the sensor readings. Studies [13] show that in particular the Domingo-Ferrer PH is very vulnerable to known plaintext attacks. The both other schemes are immune to this kind of attack.

2) Active attacks: This kind of attack assumes that the adversary is able to interfere the communication, i.e. to catch, destroy, modify, and send packets. As we will see, such an attack is the most dangerous threat against the CDA approaches for WSNs.

a) Replay attacks: A very obvious attack is the replay attack, i.e. the malicious resending of previously sent packets. In WSNs this means a regular packet is resent at wrong time. Please consider a movement detection scenario. A trespasser can keep sending the 'no movement' signal while he is moving in the protected area. From our knowledge only the CMT resists this kind of attack, because it applies a new key for each message.

b) Malleability: The idea of this very dangerous attack is to change the content of a valid encrypted packet without leaving marks. For CMT and ECEG it is possible to alter the content of an encrypted packet, without knowing the plaintext. An adversary can alter the encrypted values of CMT or ECEG by adding natural numbers or multiples of the generator point, respectively. Consider an adversary wants to increase the value of an encrypted ECEG value by 10. Since the generator G is part of the public key, he could add $10 \cdot G$ to the original encrypted value:

$$[kG, kH + mG] + [0, 10G] = [kG, kH + (m + 10)G)]$$

c) Unauthorized Aggregation: A threat of maliciously aggregated proper ciphertexts to a new valid but bad ciphertext. Similar to malleability. In case an adversary knows one ciphertext, he can use this packet as summand to add it, or any multiple of it, to any ciphertext without knowing its plaintext. DF and ECEG are vulnerable to this attack. CMT has a protection, because it uses a unique key for each message and expects exactly that key as part of the aggregate.

d) Forge packets: Actually, in case of ECEG there is no reason to alter existing encrypted data, since the pure application of ECEG with public keys allows everyone to create own ciphertexts with desired content. DFPH and CMT are resistant to this kind of weakness because the keys required for the encryption process are kept hidden.

3) Physical attacks: This group of threats includes all kinds of physical attacks against the node. Obviously, one could disable a node, but this would not implicitly be a threat against the security. A serious threat is the capturing of nodes. The access to the flash and memory may reveal key information that can compromise the entire network. In particular, symmetric encryption schemes that use the same key on every node are vulnerable. An example for such a scheme is the DFPH. Public key approaches like ECEG, as well as algorithms that can use different keys for each node like CMT, are resistant to the node capture attack.

TABLE I
COMPARISON OF CDA ALGORITHMS

	DF	CMT	ECEG
Ciphertext size	$-$	$+$	*o*
Comp. effort encryption	o	$+$	$-$
Comp. effort decryption	o	$-$	$--$
Comp. effort aggregation	o	$++$	$-$
Resistance regarding			
Indistinguishability	$++$	$++$	$++$
Chosen plaintext attacks	$-$	$+$	$++$
Replay attacks	$--$	$++$	$--$
Malleability	$+$	$--$	$--$
Malicious Aggregation	$-$	$+$	$--$
Forged packets	$++$	$+$	$--$
Captured sensors	$--$	$+$	$++$

B. Comparison of CDA approaches

Table I shows a brief evaluation of the described CDA algorithms regarding the set of properties and the described attack scenarios. Indeed, such an overview cannot deliver an satisfying assessment for every situation and parameter combination. For example the ciphertext size of CMT is considered as positive. But the positive assessment is not justified anymore in case where many not responding ids must be transmitted.

Another controversial point is the computation effort for ECEG. Because ECC software implementations are known to be quite slow it is assessed with '-'. However, executed on hardware accelerators ECC is very fast. More, the power consumed by the computation is even smaller than required for the transmission of the encrypted data packet. Thus, in this case the computation costs can be neglected[11].

Actually, in many application scenarios not all properties must be perfectly fulfilled. In case only a simple encryption is wanted and an active attack, which is connected with considerable expenses, is not a probable threat, all three algorithms are reasonable. In such a case side constraints could favor one algorithm or another. For example, if ECC is already part of the WSN, maybe for the key exchange protocol, ECEG is very reasonable. If malleability protection is important, DF should be selected. Of course, the security issues of every algorithm must always be kept in mind.

V. Construction of more resilient CDA methods

The previous considerations show that each of the three described CDA algorithms has its individual beneficial properties. But there are many possible attacks and none of the presented CDA algorithms provides all desirable protection features. Based on these results we discuss two general approaches: first, the combination of CDA algorithms with a more secure cryptographic system, and second, engineering of mechanism for particular properties. We focus on the CTM approach, because according to Table I it is the most promising algorithm. Its two weaknesses are the malleability and the ID-problem.

A. Combination of CDA algorithms

We showed that every PH scheme has its security issues. However, Table I shows that each protection goal is satisfied by at least one scheme. One idea to cope with the known issues is to combine two or more PH algorithms, i.e. perform cascaded encryptions:

$$E_2(E_1(a)) \oplus E_2(E_1(b)) = E_2(E_1(a+b))$$

Such a chain has some requirements on the encryption algorithms: both encryption schemes must be additive PH, and the ranges of results of inner encryption E_1 must fit to the domain of E_2. Usually E_1 is a function: $E_1 : \mathbb{Z}_n \rightarrow \mathbb{Z}_n$, while $E_2 : \mathbb{Z}_n \rightarrow \mathbb{C}$, with $\mathbb{C}$ as domain of the ciphertext. From the described PH schemes CMT is very suitable as E_1 function. Both, ECEG and DF can use CMTs results. As an example we demonstrate the combination of CMT and DFPH. In the previous section we described that CMTs only security weakness is the malleability, i.e. one can modify the content of a ciphertext without knowing the plaintext. Exactly that property is a strength of DF. Without the knowledge of the secret key one cannot modify the content of a single packet. Hence, this approach should result in the most secure CDA. The advantage of this combination is that the aggregation requires exactly the same effort as the standalone DF. Since most security concerns are already covered by CMT, the DF parameters, especially d, do not need to be too big. But, with both encryption methods, we get the technical problems of both approaches. With $d > 1$ the encrypted message size increases and there is still the id issue to indicate not responding nodes.

CMT + DF algorithm	
Parameter:	*public key: large integer* M, $d \geq 2$ *secret key:* g *that divides* M; r *so that* r^{-1} *exists in* $\mathbb{Z}_M$
Encryption:	*randomly generated keystream* $k \in [0, M-1]$ $e_1 = (k+m) \bmod M$
split	e_1 into d parts $m_1..m_d$ that $\sum_{i=1}^{d}(m_i) \bmod g = e_1$
$C = [c_1, .., c_d] = [m_1 r \bmod M, m_2 r^2 \bmod M, .., m_d r^d \bmod M]$	
Aggregation:	scalar addition modulo M (like DFPH)
Decryption:	$d_1 = (c_1 r^{-1} + ... + c_d r^{-d}) \bmod g$ $m = (d_1 - k) \bmod M$ where k is the sum of aggregated key streams

B. Mechanisms for particular problems

Actually in most cases it is not necessary to apply two complete encryption algorithms. It is usually much more efficient to apply selected mechanisms, protocols, or algorithms in order to achieve a particular property.

1) Message authentication: For example, beside the id-problem, the CMT algorithm's biggest issue is the malleability. It allows an adversary to easily add an integer to an encrypted value without knowing the plaintext or the key. For end-to-end communication digital signature schemes are the standard solution. The two-step algorithms (asymmetric encryption of a cryptographic hash value) provide three desired goals of secure communication: message integrity, authentication, and non-repudiation. The latter is not necessarily required in our WSN scenario where only the sink node must prove the origin of the message. Thus one can apply symmetric encryption of the hash value, where both sensor and sink node use the same key. This approach is known as message authentication code (MAC). We are now looking for a MAC that, beside message integrity and authentication, provides the additive homomorphic property: $MAC(a+b) = MAC(a) \oplus MAC(b)$.
Actually, such a property is not a part of standard MAC schemes, since it implies malleability. With known valid messages a and b and corresponding MACs, it is possible to forge a new packet a+b. One solution for this issue is a limited life time of a MAC, so that an adversary cannot obtain two MACs that can be combined into a malicious one. The idea is to create a MAC based on a nonce, the sensor id, the value, and the secret key: $MAC((Nonce, SensorID, Value), Key)$
The nonce can be taken from a key stream that provides a unique key for each message, an embedded time stamp, or a challenge-response system that requires the sensor to encrypt a specific nonce, provided and expected by the sink node.
The key stream approach is related to the CMT encryption. A time stamp rises the problem of time synchronization. If one sensor delivers the wrong time, the whole aggregated signature is invalid. The third idea is not connected with such problems but requires a very secure hashing and encryption mechanism since it can be assumed that an attacker knows the nonce. Anyway, the hash function and encryption mechanism is the biggest issue. It must be provided that it is not possible to change nonce or sensor id of a valid MAC to specific values without knowing the encryption key. Additionally, it must be provided that values cannot be changed unnoticed.

2) Efficient ID transmission: The ID-issue, which has been described as major disadvantage of the CMT scheme, is probably a problem for any message authentication mechanism, too. If it should be verified that a value has been aggregated properly, it must be known that the value is actually part of the aggregate.

In [2] was shown that in an exemplary 3-tree of height 7 with 3280 sensor nodes the application of CMT improves the bandwidth performance by a factor of about 5 compared to an ETE approach. I.e. the number of transmitted bits is reduced to one fifth. The performance gain becomes worse if

the number of not responding nodes rises, since the ids of not responding nodes are attached. The simple attachment of not responding nodes is very efficient if their number is not high, but in case of many disabled nodes the overhead is crucial. For the described scenario in the worst case, where every second node is not responding, more than 100 thousand additional bits are required. In such a situation it is much more efficient to assign the status of every sensor in a bit array. In our scenario it needs less than 20 thousand bits for the id information. The size is independent on the error rate, thus in case of only several non-responding nodes it is not really beneficial.

Driven by the question for an optimal encoding, we applied combinatorics to determine the minimum number of bits required for reporting e randomly distributed nodes out of n sensors. It can be written as

$$log_2 \binom{n}{e} = log_2 \left(\frac{n!}{e!(n-e)!} \right) = \sum_{i=n-e+1}^{n} log_2 i - \sum_{i=1}^{e} log_2 i$$

The result represents a lower bound of bits. For $e = 1$ it is exactly $log_2 n$, i.e. the id of the one node is transmitted. Figure 1 shows a graphical representation of the number of required bits as function of responding nodes. It is a notion to use situation depended either the list of ids or node array, whichever is better. Whether the required transformations of the coding is worth the efforts and which other coding schemes are feasible must be faced in further work.

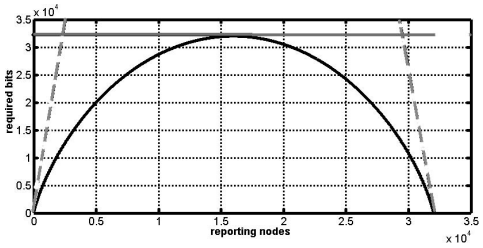

Fig. 1. Required bits for reporting status of 32000 nodes with different approaches. The black solid curve is the minimum number corresponding to the equation. The dashed lines at the left and the right represent the number of bits needed for reporting the responding and not responding nodes, respectively. The gray horizontal line is the amount of bits if the complete bit array is used.

VI. Summary and Outlook

In this paper we presented the concealed data aggregation (CDA) as secure version of the in-network aggregation (INA) for wireless sensor networks. In contrast to traditional secure INA approaches, which decrypt and re-encrypt the values on every aggregation node, CDA provides end-to-end security. I.e. even though the sensed data are encrypted on the sensor nodes and not decrypted before the sink node, they can be aggregated on the intermediate nodes. We described three algorithms that allow additive CDA that may suit to the WSN scenario. On the discussion of algorithms we emphasized the security properties, i.e. the resilience against specific attacks. We discovered that none of the described algorithms provides all the desirable security goals. Despite this, it turned out that the key stream based CMT approach is the most promising one. It is very efficient and the most secure algorithm from the discussed ones. However, it still has its weaknesses.

To cope with the problems we propose two approaches. The first approach combines two algorithms so that weaknesses of one algorithm are covered by the strengths of the other one. As example we demonstrated how to combine CMT with the Domingo-Ferrer algorithm. The resulting CMT-DF algorithm promises high security for reasonable additional effort. In our second approach we propose to engineer mechanisms to counter the particular weaknesses. We propose homomorphic message authentication code to overcome the malleability problems. Finally we discussed the required effort for the transmission of the sensor ids that are part of the aggregate. This information is substantial for CMT and probably any homomorphic MAC.

Obviously there are still many open questions. Next we will face the construction of a working homomorphic MAC that ensures the authentication of the aggregate, preferable in combination with the key stream based CMT algorithm. It is necessary to look for efficient sophisticated coding methods to reduce the network overhead for the transmissions of the IDs. And the security of the algorithms and constructions must be further analyzed. Additionally, the effectiveness and efficiency of the approaches must be verified—deployed on WSNs.

References

[1] Jim Adler et.al. Computational details of the votehere homomorphic election, 2000. http://www.votehere.net.

[2] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. Effcient aggregation of encrypted data in wireless sensor networks. In *MobiQuitous*, pages 109–117. IEEE Computer Society, 2005.

[3] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *ISC '02: Proceedings of the 5th International Conference on Information Security*, 2002.

[4] Joao Girao, Dirk Westhoff, and Markus Schneider. Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks. In *IEEE International Conference on Communications*, 2005.

[5] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[6] Einar Mykletun, Joao Girao, and Dirk Westhoff. Public key based cryptoschemes for data concealment in wireless sensor networks. In *IEEE International Conference on Communications*. ICC2006, 2006.

[7] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *ACM Conference on Computer and Communications Security*, pages 59–66, 1998.

[8] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. *Lecture Notes in Computer Science*, 1403:308–??, 1998.

[9] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Lecture Notes in Computer Science*, 1592, 1999.

[10] Pascal Paillier. Trapdooring discrete logarithms on elliptic curves over rings. *Lecture Notes in Computer Science*, 1976:573–??, 2000.

[11] Krzysztof Piotrowski, Peter Langendoerfer, and Steffen Peter. How public key cryptography influences wireless sensor node lifetime. In *Proceedings of the 4rd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2006*, 2006.

[12] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In R. DeMillo, D. Dobkin, A. Jones, and R. Lipton, editors, *Foundations of Secure Computation*, pages 169–180. Academic Press, 1978.

[13] David Wagner. Cryptanalysis of an algebraic privacy homomorphism. In *Information Security, 6th International Conference, ISC 2003, Bristol, UK, Proceedings*, pages 234–239, 2003.