1-2016

Privacy-preserving and verifiable data aggregation

Ngoc Hieu TRAN
Singapore Management University, nhtran.2013@phdis.smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016
A. Mathur and A. Roychoudhury (Eds.)

doi:10.3233/978-1-61499-617-0-115

Privacy-Preserving and Verifiable Data Aggregation

Hieu N TRAN [a], Robert H DENG [a] and HweeHwa PANG [a]
[a] *School of Information Systems, Singapore Management University*

Abstract. There are several recent research studies on privacy-preserving aggregation of time series data, where an aggregator computes an aggregation of multiple users' data without learning each individual's private input value. However, none of the existing schemes allows the aggregation result to be verified for integrity. In this paper, we present a new data aggregation scheme that protects user privacy as well as integrity of the aggregation. Towards this end, we first propose an aggregate signature scheme in a multi-user setting without using bilinear maps. We then extend the aggregate signature scheme into a solution for privacy-preserving and verifiable data aggregation. The solution allows multiple users to periodically send encrypted data to an untrusted aggregator such that the latter is able to compute the sum of the input data values and verify its integrity, without learning any other information. A formal security analysis shows that the solution is semantically secure and unforgeable.

Keywords. data aggregation, data privacy, verifiable computation, aggregate signature

1. Introduction

Data aggregation, in which an aggregator computes aggregated results over multiple sources, has been widely used in real-world applications such as network coding, smart grid, mobile sensing and cloud services. In order to prevent leakage of sensitive information, the aggregator should be able to perform aggregation operations such as sum, average, variance on users' data without learning their individual values. Symmetric key homomorphic encryption schemes, which are normally based on one-time pad encryption, could be applied to protect the confidentiality of user data [1]. However, such schemes assume that the aggregator is fully trusted since each user shares a secret key with the aggregator. Therefore, these schemes protect data confidentiality against outsiders, but they do not protect users' data privacy against the aggregator.

Recently, several research studies have addressed the privacy-preserving data aggregation problem under the assumption of an untrusted aggregator [2,3]. Shi et al. proposed a construction that allows the untrusted aggregator to compute the sum of time series data [2]. However, the aggregator must use brute-force search or Pollard's lambda method [4] to decrypt the sum, which is not efficient for applications requiring a large plaintext space. Li et al. [3] proposed an efficient aggregation scheme based on a novel key management technique, where only computation of key hashes is needed for encrypting data by users and for decrypting the sum of data values by the aggregator.

In many critical control and monitoring systems, ensuring the integrity of aggregation results is extremely important. In such systems, any invalid aggregation result due to tampering by malicious attackers must be detected and rejected by the aggregator. There are two generic mechanisms for verifying the integrity of aggregation results aggregate message authentication codes (MACs) [5,6] and aggregate signatures [7,8,9,10,11,12]. The notion of aggregate MAC was first proposed by Katz et al. [5]. Agrawal et al. [13] proposed a homomorphic MAC scheme, a generalization of aggregate MAC, that allows a network node to compute a MAC of combined code words in networking coding. Aggregate MAC and homomorphic MAC require shared secret keys between the data generators and aggregators. In contrast, aggregate signature schemes enable anyone with the correct public keys to verify the integrity of aggregation results. However, most of the existing aggregate signature schemes [7,8,9,10,11,12] are constructed using bilinear maps and are not suitable for resource-constrained system settings such as sensor networks.

Our Contribution. Our goal is to propose an efficient construction for privacy-preserving and verifiable data aggregation. The main contributions of this paper are as follows:

- We first introduce an aggregate signature scheme for public verification of the integrity of aggregated data in a multi-user setting. In this scheme, each user sends his data and a signature to an aggregator, where the signature is generated on the data using the user's secret key. The aggregator combines all the users' data and signatures into an aggregated result and an aggregated signature, respectively, and checks the validity of the result using public system parameters.
- Extending the aggregate signature scheme, we further propose a privacy-preserving and verifiable data aggregation (PVDA) scheme. PVDA allows multiple users to periodically send encrypted data to an untrusted aggregator such that the latter is able to compute the sum of the data values and verify its integrity but not learn any other information. Our formal security analysis shows that the solution is semantically secure and unforgeable.

2. Related Work

We survey prior work on privacy-preserving and verifiable data aggregation, aggregate MAC and aggregate signature.

Privacy-Preserving Data Aggregation Many protocols have been proposed for privacy-preserving data aggregation in wireless sensor network (WSN) or mobile sensing [1,3,14,15,16,17], without considering the integrity of aggregation result. Shi et al. [2] proposed a construction that allows a group of users to send encrypted data to an untrusted aggregator. The construction is not efficient for applications requiring a large plaintext space because the aggregator must use brute-force search to decrypt a sum. Li et al. [3] extended the construction in [2] using a new key management technique to ensure that the aggregator could only obtain the sum of the users' private data but not the private data of each user. Our PVDA construction solution is motivated by [2,3], but provide verifiable assurance in addition to data privacy.

Aggregate MAC and Aggregate Signature Aggregate MACs [5,13] allow verification of the integrity of aggregation data in private setting. Recently, Catalano and Fiore [18] proposed homomorphic MAC schemes that support a set of functionalities (i.e., polynomially-bounded arithmetic circuits); however, in their construction, the verifier possesses the user's secret key and hence is able to access users' private data. Aggregate signatures [8,9,10,11,12] also allow verification of the integrity of aggregation data but in public setting. Most of the existing aggregate signature (or more generally homomorphic signature) schemes are constructed on bilinear maps and unsuitable for resource-constrained applications.

Verifiable and Privacy-Preserving Data Aggregation Lin et al. [19] proposed a multi-dimensional privacy preserving data aggregation scheme for WSN. This scheme adapts the super-increasing sequences and permutation techniques from [1] for data aggregation. However, the scheme assumes that the aggregator is fully trusted. Papadopoulos et al. [20] presented a scheme that utilizes a combination of homomorphic encryption and secret sharing. Although their scheme works efficiently on time series data, the data privacy of each node may be breached if the querier colludes with the aggregator, since the querier has the secret key of every node.

3. A New Aggregate Signature Scheme

3.1. Overview

Our system model consists of three types of entities: a set of n users $\{u_1,...,u_n\}$, an aggregator, and a key generation center. The key generation center creates a public parameter $\mathscr{S}$, as well as a key pair sk_i and pk_i for every user u_i. We assume that key pairs $(\mathsf{sk}_i, \mathsf{pk}_i)$ for all $i = 1,...,n$ are distributed to the corresponding users via a secure channel at system initialization.

In each time period, each user u_i generates a private datum $m_i \in \mathbb{Z}_p$ where p is a large prime number. The aggregator is a powerful base station that performs data aggregation operations such as $\mathtt{sum}(M) = m_1 + ... + m_n$, where $M = (m_1,...,m_n)$ is a vector of all the users' private data.

We assume that the aggregator is honest but curious. He is trusted to perform aggregation operations, but may attempt to discover individual users' private data. We also assume that users follow the correct aggregation process, do not trust each other and are curious as well. Some curious users may collude with the aggregator by revealing their secret keys in order to deduce additional information about the remaining users.

3.2. Definition

An aggregate signature scheme is a tuple of three probabilistic polynomial-time (PPT) algorithms (KeyGen, Sign, Verify) as follows:

- $\mathsf{KeyGen}(1^\lambda)$ algorithm is executed by the key generation center. It takes as input a security parameter λ and outputs a pair of public key pk_i and secret key sk_i for each user u_i along with a set of random values $\mathscr{S}$ as public parameter.

- $\mathsf{Sign}(\mathsf{sk}_i, m_i, t)$ algorithm is executed by each user u_i. It takes as input a secret key sk_i, datum m_i and time period t, and outputs a signature $\tau_i \in \mathbb{Z}_p$. User u_i then sends (m_i, τ_i) to the aggregator.
- $\mathsf{Verify}(\mathsf{pk}, \mathscr{S}, M, T, t)$ algorithm is executed by the aggregator. It takes as input a vector of public keys $\mathsf{pk} = (\mathsf{pk}_1, ..., \mathsf{pk}_n)$, the set $\mathscr{S}$, a vector of data $M = (m_1, ..., m_n)$, a vector of signatures $T = (\tau_1, ..., \tau_n)$ and a time period t. It outputs '1' *(accept)* or '0' *(reject)*.

Security. We allow the adversary to request a vector of signatures T for an arbitrary vector of data M along with period time t of his choice. Note that set $\mathscr{S}$ is a public parameter, hence anyone can compute and verify the integrity of an aggregation result. The adversary breaks the aggregate signature scheme if he is able to output a valid triple (M, T, t) where either period time t is new, or he has not previously requested signatures on the vector of data M. We define the security of our scheme in terms of the following game between a challenger and an adversary $\mathscr{A}$.

Game 1 *Let $\mathscr{H} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ denote an aggregate signature scheme.*

- *Setup: The challenger generates a set of random values $\mathscr{S}$ and n key pairs $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_{i=1}^n$, then sends $\mathscr{S}$ and $\mathsf{pk} = (\mathsf{pk}_1, ..., \mathsf{pk}_n)$ to the adversary.*
- *Queries: The adversary submits signature queries, each of the form (M_j, t_j) where $M_j = (m_{1,j}, ..., m_{n,j}) \in \mathbb{Z}_p^n$ and $t_j \in \{0,1\}^*$. We require the time periods t_j submitted by the adversary to be distinct. The challenger performs the following:*

  for all $i = 1, ..., n$ ***do***
  $\tau_{i,j} \leftarrow \mathsf{Sign}(\mathsf{sk}_i, m_{i,j}, t_j)$
  send $T_j = (\tau_{1,j}, ..., \tau_{n,j})$ *to* $\mathscr{A}$
- *Output: The adversary outputs a vector of data M^*, a vector of signatures T^* and a time period t^*. The adversary wins the security game if $1 \leftarrow \mathsf{Verify}(\mathsf{pk}, \mathscr{S}, M^*, T^*, t^*)$, and one of the following conditions hold:*

  * *Type 1 forgery: $t^* \neq t_j$ for all j.*
  * *Type 2 forgery: $t^* = t_j$ for some j, and $M^* \neq M_j$.*

The advantage $\mathsf{uf\text{-}adv}[\mathscr{A}, \mathscr{H}]$ of adversary $\mathscr{A}$ with the aggregate signature scheme $\mathscr{H}$ is defined as the probability that $\mathscr{A}$ wins Game 1.

Definition 1 *The aggregate signature scheme $\mathscr{H}$ is existentially unforgeable under an adaptive chosen-message attack if, for all PPT adversaries $\mathscr{A}$, the advantage $\mathsf{uf\text{-}adv}[\mathscr{A}, \mathscr{H}]$ is negligible.*

3.3. Construction

Let $\mathbb{G}$ be a cyclic group of prime order p on which the discrete logarithm problem is hard. Let $M = (m_1, ..., m_n) \in \mathbb{Z}_p^n$ denote a vector of n users' data values, where $m_i < p$ for $i = 1, ..., n$ and $\sum_{i=1}^n m_i \leq p$. Let $f : \mathbb{Z}_p \times \{0,1\}^* \rightarrow \mathbb{Z}_p$ be a pseudo-random function (PRF). The three algorithms (KeyGen, Sign, Verify) are given as follows.

- $\mathsf{KeyGen}(1^\lambda)$: The key generation center chooses a random generator $g \in \mathbb{G}$, generates a set of $n\alpha$ random values $\mathscr{S} = (s_1, ..., s_{n\alpha}) \in \mathbb{Z}_p^{n\alpha}$ and divides them into

n disjoint subsets $\mathscr{S}_i$ such that each subset $\mathscr{S}_i$ has α values, $\mathscr{S} = \bigcup_{i=1}^{n} \mathscr{S}_i$ and $\forall i \neq j, \mathscr{S}_i \bigcap \mathscr{S}_j = \emptyset$. The key generation center assigns $\mathscr{S}_i$ to user u_i and $\mathscr{S}$ to the aggregator. Each user u_i then generates a random integer a_i and computes $h_i = g^{a_i}$. Each user u_i has a secret key $\mathsf{sk}_i = \langle \mathscr{S}_i, a_i \rangle$ and a public key $\mathsf{pk}_i = \langle g, h_i \rangle$.

- $\mathsf{Sign}(\mathsf{sk}_i, m_i, t)$: Given secret key sk_i and datum m_i for period time t, user u_i computes a signature $\tau_i = a_i \times m_i + \mathsf{F}_{\mathscr{S}_i}(t) \bmod p$, where $\mathsf{F}_{\mathscr{S}_i}(t) = \sum_{s \in \mathscr{S}_i} f_s(t) \bmod p$, then sends (m_i, τ_i) to the aggregator.
- $\mathsf{Verify}(\mathsf{pk}, \mathscr{S}, M, T, t)$: For each time period t, the aggregator computes an aggregate signature $\mathscr{T} = \sum_{i=1}^{n} \tau_i = \sum_{i=1}^{n} (a_i \times m_i + \mathsf{F}_{\mathscr{S}_i}(t)) \bmod p$. The aggregator then verifies the integrity of the aggregate signature $\mathscr{T}$ using public keys $\mathsf{pk} = (\mathsf{pk}_1, ..., \mathsf{pk}_n)$ and $\mathscr{S} = \{s_1, ..., s_{n\alpha}\}$ as follows:
  - $Y = \prod_{i=1}^{n} h_i^{m_i} \in \mathbb{G}$.
  - If $g^{\mathscr{T}} = g^{\sum_{s \in \mathscr{S}} f_s(t)} \times Y$, output '1' *(accept)*; otherwise output '0' *(reject)*.

Security. Adversary $\mathscr{A}$ is able to succeed in a forgery attack and win the security Game 1 if he knows the secret key of any of the n users. However, his chance of guessing any user's secret key is negligible, as Lemma 1 shows. We prove security assuming F is a secure PRF, and the discrete logarithm assumption holds in $\mathbb{G}$. We refer to [4] for a definition of the PRF and discrete logarithm security games. The proofs are given in the full version of this paper.

Lemma 1 *In our aggregate signature scheme, the probability of successfully discovering any user's private key through brute-force guessing is negligible.*

Theorem 1 *The aggregate signature scheme is existentially unforgeable under an adaptive chosen-message attack, assuming F is a secure PRF and the discrete logarithm problem is hard.*

4. Privacy-Preserving and Verifiable Data Aggregation

The aggregate signature scheme introduced in the previous section does not maintain *confidentiality* of users' data. We now propose a privacy-preserving and verifiable data aggregation (PVDA) scheme that extends the aggregate signature scheme.

4.1. Definition

PVDA comprises four algorithms (KeyGen, Enc, Sign, AggDec) as follows:

- $\mathsf{KeyGen}(1^{\lambda})$ algorithm remains the same as in Section 3.2.
- $\mathsf{Enc}(\mathsf{sk}_i, m_i, t)$ algorithm is executed by each user u_i. It takes as input secret key sk_i, datum m_i and time period t, and outputs a ciphertext $c_i \in \mathbb{Z}_p$.
- $\mathsf{Sign}(\mathsf{sk}_i, c_i, t)$ algorithm is as in Section 3, except that m_i is replaced by ciphertext c_i. Each user sends (c_i, τ_i) to the aggregator where $\tau_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, c_i, t)$.
- $\mathsf{AggDec}(\mathsf{pk}, \mathscr{S}, C, T, t)$ algorithm is executed by the aggregator. It takes as input public keys $\mathsf{PK} = (\mathsf{pk}_1, ..., \mathsf{pk}_n)$, the set $\mathscr{S}$, a vector of ciphertexts $C = (c_1, ..., c_n)$, a vector of signatures $T = (\tau_1, ..., \tau_n)$ and time period t. It outputs an aggregation result $\mathscr{M} = m_1 + ... + m_n$ if the Verify algorithm as in Section 4.2 outputs '1'; otherwise it outputs error $\perp$.

Security. The security notion for PVDA includes *semantic security* and *unforgeability*. Let $\mathscr{P} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Sign}, \mathsf{AggDec})$ denote the PVDA scheme.

Game 2 *Semantic security of the PVDA scheme is defined by the following game between a challenger and an adversary $\mathscr{A}$:*

- *Setup: The challenger generates a set $\mathscr{S}$ of $n\alpha$ random values and n key pairs (sk_i, pk_i) for $i = 1,...,n$. The challenger gives the public parameters $\langle \mathscr{S}, pk = (pk_1,...,pk_n)\rangle$ to the adversary. The former also initializes a list $L = \emptyset$ for tracking the queries from $\mathscr{A}$.*
- *Queries: The adversary adaptively queries the challenger for encryption. Each query states $m \in \mathbb{Z}_p$, index i of user u_i and time period t. The query is rejected if the tuple (i,t) exists in L; otherwise, the challenger computes $c \leftarrow \mathsf{Enc}(sk_i, m, t)$ and $\tau \leftarrow \mathsf{Sign}(sk_i, c, t)$, sends (c, τ) to $\mathscr{A}$ and inserts (i,t) into L.*
- *Challenge: The adversary $\mathscr{A}$ submits m_0, $m_1 \in \mathbb{Z}_p$ along with index i^* and time period t^* to the challenger; L must not already contain the tuple (i^*, t^*). The challenger computes $c^* \leftarrow \mathsf{Enc}(sk_{i^*}, m_b, t^*)$, $\tau^* \leftarrow \mathsf{Sign}(sk_{i^*}, c^*, t^*)$ under a random bit $b \in \{0,1\}$, and sends (c^*, τ^*) to adversary $\mathscr{A}$.*
- *Output: Adversary $\mathscr{A}$ outputs b', representing his guess for b, and wins the game if $b' = b$.*

The advantage $\mathsf{ss\text{-}adv}[\mathscr{A}, \mathscr{P}]$ of adversary $\mathscr{A}$ with respect to $\mathscr{P}$ is defined as $|\Pr[b = b'] - \frac{1}{2}|$, where the probability is taken over the random bit used by the challenger and the adversary $\mathscr{A}$.

Definition 2 *The PVDA scheme $\mathscr{P}$ is semantically secure if, for all PPT adversary $\mathscr{A}$, the advantage $\mathsf{ss\text{-}adv}[\mathscr{A}, \mathscr{P}]$ is negligible.*

Game 3 *Unforgeability of the PVDA scheme is defined by the following game between a challenger and an adversary $\mathscr{A}$:*

- *Setup: The challenger generates a set $\mathscr{S}$ of $n\alpha$ random values and n key pairs (SK_i, PK_i) for $i = 1,...,n$. The challenger gives public parameters $\langle \mathscr{S}, PK_1,..., PK_n\rangle$ to the adversary.*
- *Queries: The adversary adaptively submits (M_j, t_j) to the challenger as in Game 1. The challenger responds with $(c_{1,j}, \tau_{1,j}),..., (c_{n,j}, \tau_{n,j})$ for each query, where $c_{i,j}$ is the ciphertext of $m_{i,j}$, and $\tau_{i,j}$ is the signature of $c_{i,j}$ for $i = 1,...,n$. We require all the time periods t_j's in the queries to be distinct.*
- *Output: The adversary outputs the set of n pairs (c_1^*, τ_1^*) ,..., (c_n^*, τ_n^*) for $M^* = (m_1^*,..., m_n^*)$ and time period t^* such that either t^* is a new time period or M^* has not been queried before. The adversary wins the game if the AggDec algorithm outputs $\mathscr{M}^* = \sum_{i=1}^n m_i^*$.*

The advantage $\mathsf{uf\text{-}adv}[\mathscr{A}, \mathscr{P}]$ of adversary $\mathscr{A}$ with respect to $\mathscr{P}$ is defined as the probability that $\mathscr{A}$ wins Game 3.

Definition 3 *Our PVDA scheme $\mathscr{P}$ is unforgeable if, for all PPT adversary $\mathscr{A}$, the advantage $\mathsf{uf\text{-}adv}[\mathscr{A}, \mathscr{P}]$ is negligible.*

4.2. Construction

In order to protect the privacy of users' data while providing verifiability of aggregation results, we use the technique of Li et al. [3] in encryption. However, PVDA is more practical because Li et al.'s scheme does not provide verifiability. The four algorithms (KeyGen, Enc, Sign, AggDec) in PVDA are given below.

- $\mathsf{KeyGen}(1^\lambda)$: The same as in the aggregate signature scheme.
- $\mathsf{Enc}(\mathsf{sk}_i, m_i, t)$: User u_i generates encryption keys $k_i = \mathsf{F}_{\mathscr{S}_i}(t\|1)$ and $k_i' = \mathsf{F}_{\mathscr{S}_i}(t\|2)$, where $\|$ denotes concatenation. Next, the user encrypts his datum m_i by computing a ciphertext $c_i = m_i + k_i \bmod p$.
- $\mathsf{Sign}(\mathsf{sk}_i, c_i, t)$: User u_i computes a signature τ_i on the output c_i of the Enc algorithm using his private value a_i and encryption key k_i' by computing $\tau_i = a_i \times c_i + k_i' \bmod p$. User u_i then sends (c_i, τ_i) to the aggregator.
- $\mathsf{AggDec}(\mathsf{pk}, \mathscr{S}, C, T, t)$: In each time period t, the aggregator generates decryption keys $k_0 = \mathsf{F}_{\mathscr{S}}(t\|1)$ and $k_0' = \mathsf{F}_{\mathscr{S}}(t\|2)$. Upon receiving $C = (c_1, ..., c_n)$ and $T = (\tau_1, ..., \tau_n)$ from all the n users, the aggregator computes:

$$\mathscr{T} = \sum_{i=1}^{n} \tau_i \bmod p, \qquad \mathscr{X} = \sum_{i=1}^{n} c_i \bmod p, \qquad \mathscr{Y} = \prod_{i=1}^{n} h_i^{c_i}$$

  The aggregation signature $\mathscr{T}$ is valid if and only if $g^{\mathscr{T}} = g^{k_0'} \times \mathscr{Y}$. The aggregator outputs aggregation result $\mathscr{M} = \sum_{i=1}^{n} m_i = \mathscr{X} - k_0 \bmod p$ if signature $\mathscr{T}$ is valid; otherwise, it returns error $\perp$.

Security. We now formally state and prove the semantic security and unforgeability assurances of PVDA. We note that Lemma 1 applies to PVDA. The proofs of the security theorems are given in the full version of this paper.

Theorem 2 *The PVDA scheme $\mathscr{P}$ is semantically secure if, for all PPT adversaries $\mathscr{A}$, the advantage $\mathsf{ss\text{-}adv}[\mathscr{A}, \mathscr{P}]$ is negligible assuming F is a secure PRF.*

Theorem 3 *The PVDA scheme $\mathscr{P}$ is unforgeable if, for all PPT adversaries $\mathscr{A}$, the advantage $\mathsf{uf\text{-}adv}[\mathscr{A}, \mathscr{P}]$ of adversary $\mathscr{A}$ is negligible assuming F is a secure PRF and the discrete logarithm problem is hard.*

5. Conclusions

In this paper, we presented an aggregate signature scheme that is suitable for data aggregation in a multi-user setting. In the scheme, each user computes a signature on his data using a secret key. An aggregator computes the sum of the users' data and combines their signatures into a single aggregate signature. Using public system parameters and the aggregate signature, the aggregator is able to verify the integrity of the sum. To protect the users' data privacy, we further extended the aggregate signature scheme into a privacy-preserving and verifiable data aggregation (PVDA) scheme for time series data in the same multi-user setting. We formally proved that the PVDA scheme is semantically secure and existentially unforgeable. Compared to existing alternatives in the literature, our PVDA scheme offers two key advantages in assuming an untrusted aggregator, and being computationally efficient as no bilinear maps are used.

References

[1] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *2nd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous)*, pp. 109–117, 2005.

[2] E. Shi, T. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2011.

[3] Q. Li and G. Cao, "Efficient and privacy-preserving data aggregation in mobile sensing," in *20th IEEE International Conference on Network Protocols, ICNP*, pp. 1–10, 2012.

[4] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[5] J. Katz and A. Y. Lindell, "Aggregate message authentication codes," in *Topics in Cryptology - CT-RSAs*, pp. 155–169, 2008.

[6] A. C. Chan and C. Castelluccia, "On the (im)possibility of aggregate message authentication codes," in *2008 IEEE International Symposium on Information Theory, ISIT*, pp. 235–239, 2008.

[7] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology - EUROCRYPT*, pp. 416–432, 2003.

[8] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Public Key Cryptography - PKC*, pp. 257–273, 2006.

[9] D. Boneh, D. M. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, pp. 68–87, 2009.

[10] N. Attrapadung and B. Libert, "Homomorphic network coding signatures in the standard model," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, pp. 17–34, 2011.

[11] D. Catalano, D. Fiore, and B. Warinschi, "Efficient network coding signatures in the standard model," in *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pp. 680–696, 2012.

[12] D. M. Freeman, "Improved security for linearly homomorphic signatures: A generic framework," in *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pp. 697–714, 2012.

[13] S. Agrawal and D. Boneh, "Homomorphic macs: Mac-based integrity for network coding," in *Applied Cryptography and Network Security, 7th International Conference, ACNSs*, pp. 292–305, 2009.

[14] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems," in *INFOCOM 2010. 29th IEEE International Conference on Computer Communications*, pp. 758–766, 2010.

[15] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 56–60, 2008.

[16] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, "PDA: privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pp. 2045–2053, 2007.

[17] T. Jung, X. Mao, X. Li, S. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation," in *Proceedings of the IEEE INFOCOM*, pp. 2634–2642, 2013.

[18] D. Catalano and D. Fiore, "Practical homomorphic macs for arithmetic circuits," in *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pp. 336–352, 2013.

[19] X. Lin, R. Lu, and X. S. Shen, "MDPA: multidimensional privacy-preserving aggregation scheme for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 6, pp. 843–856, 2010.

[20] S. Papadopoulos, A. Kiayias, and D. Papadias, "Secure and efficient in-network processing of exact SUM queries," in *Proceedings of the 27th International Conference on Data Engineering, ICDE*, pp. 517–528, 2011.