DP^{2}AC: Distributed Privacy-Preserving Access Control in Sensor Networks

Rui Zhang, Yanchao Zhang
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Email: {rz23, yczhang}@njit.edu

Kui Ren
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Email: kren@ece.iit.edu

***Abstract*—The owner and users of a sensor network may be different, which necessitates privacy-preserving access control. On the one hand, the network owner need enforce strict access control so that the sensed data are only accessible to users willing to pay. On the other hand, users wish to protect their respective data access patterns whose disclosure may be used against their interests. This paper presents DP^{2}AC, a Distributed Privacy-Preserving Access Control scheme for sensor networks, which is the first work of its kind. Users in DP^{2}AC purchase tokens from the network owner whereby to query data from sensor nodes which will reply only after validating the tokens. The use of blind signatures in token generation ensures that tokens are publicly verifiable yet unlinkable to user identities, so privacy-preserving access control is achieved. A central component in DP^{2}AC is to prevent malicious users from reusing tokens. We propose a suite of distributed techniques for token-reuse detection (TRD) and thoroughly compare their performance with regard to TRD capability, communication overhead, storage overhead, and attack resilience. The efficacy and efficiency of DP^{2}AC are confirmed by detailed performance evaluations.**

I. Introduction

There are three ways to access data in a sensor network. First, sensor nodes can transmit their data through a base station to an external data logger which centrally handles data queries. Second, users can issue data queries through a base station which in turn forwards query results to users. Third, users can enter the sensor field to directly access data on sensor nodes without involving a base station. The third method is the only feasible option for sensor networks deployed in extreme and hazardous environments such as oceans and animal habitats, where it may be impossible or prohibitive to maintain a stable communication connection between an in-network base station and the outside network. As [1]–[7], this paper is concerned with the third data access method.

Owners and users of sensor networks may be different, which necessitates privacy-preserving access control. For example, increasing programs and projects such as ORION [8], NOPP [9] and IOOS [10] are constructing large-scale networked sensor systems to adaptively observe the earth-ocean-atmosphere system. The sensed data may be of interest to numerous users from both public and private sectors, ranging from individual users to universities, government research centers, and business companies. To compensate for operating and maintenance costs, the network owner may have to enforce strict access control so that the sensed data are accessible only by users willing to pay. There is also a growing requirement for protecting users' data access privacy [11], [12]. In particular, a user may want to keep confidential whether/when he accessed the sensed data, the data types he was interested in, or from which nodes he obtained the data, as the disclosure of such information may be used against his interest. For example, an oil company interested in the data of an ocean sensor network [8]–[10] may want to hide its network regions of interest from both the network owner and other network users that might be potential business competitors [11].

Privacy-preserving access control in sensor networks has so far received little attention, despite a rich literature (e.g., [13]–[16]) on securing sensor network communications. Related work [2]–[7] addresses access control by authenticating network users before granting them data access rights, but the privacy of users is not considered. As far as we know, SPYC [11] is the only work that takes into consideration the access privacy of network users. SPYC assumes the second data acquisition method mentioned above, i.e., that users acquire data through one or multiple base stations which may not exist in our target scenarios. It is thus a centralized solution orthogonal to our work in this paper.

In this paper, we present DP^{2}AC, a Distributed Privacy-Preserving Access Control scheme for single-owner multi-user sensor networks. In DP^{2}AC, each user interested in sensed data buys some *token*s from the network owner before entering the sensor network, who can subsequently send a query and an unspent token to any sensor node. Once validating the token, the sensor node can provide the user with an appropriate amount of requested data commensurate with the denomination of the token. Token generations involve blind signatures [17], which lead to a desirable property: the validity of each token can be verified by any sensor node, but no one, including the network owner, can tell the identity of the token holder. In this way, the network owner can prevent unauthorized access to sensed data, while users can protect their data access privacy. DP^{2}AC is a non-trivial adaptation of untraceable electronic cash systems [17]–[19] to resource-poor sensor networks.

A central issue in DP^{2}AC is detecting reused tokens. Each token in DP^{2}AC is essentially a random bit string with no relationship to user identities. Malicious users thus may

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

have financial interest in reusing tokens at different sensor nodes without concerning being caught. This would result in substantial financial losses of the network owner if there are many malicious users. The most straightforward solution for token-reuse detection (TRD) is to let each sensor node check with an in-network base station that a token was not spent and otherwise reject the data access request. To enable this, the base station need record every token submitted by sensor nodes. This centralized method can detect every token-reuse attempt, but the base station is the single of failure: once compromising the base station, malicious users can freely reuse tokens. In addition, if there are many tokens to verify, sensor nodes close to the base station would deplete their energy quickly for relaying TRD requests and results. Moreover, the base station may not exist in our target scenarios. This situation calls for distributed TRD (DTRD) schemes. In this paper, we propose a suite of DTRD techniques and thoroughly compare their performance with regard to TRD capability, communication overhead, storage overhead, and attack resilience. All these schemes rely on the collaboration of sensor nodes themselves without a single point of failure. Detailed performance evaluations confirm the efficacy and efficiency of the proposed DTRD techniques in thwarting token-reuse attempts. In particular, our techniques can detect any token-reuse attempt with probability near 1, making DP^{2}AC a very practical and trustworthy solution for sensor networks.

II. Network and Trust Models

A. Network Model

We assume a single-owner multi-user large-scale sensor network with N sensor nodes which continuously produce data of interest to many users from both pubic and private sectors besides the network owner itself. Such sensor networks are under construction or planning by many multi-sponsor programs and projects [8]–[10]. There may or may not be an in-network base station bridging the sensor network to the outside network. Our DP^{2}AC can apply to either case for its independence of base station. As in related work [2]–[7], we assume that sensor nodes know their geographical locations which can be acquired via many existing localization schemes.

B. Trust Model

Recent years have witnessed a flurry of research activities in securing sensor networks, see for example [13]–[16]. This paper focuses on privacy-preserving access control exerted on users interested in sensed data. We resort to the existing rich literature for other important issues such as key management, secure routing, broadcast authentication, and DoS mitigation.

We assume that the network owner charges users for accessing sensed data, thus enforcing strict access control. The network owner is trusted to provide the appropriate amount of data commensurate with users' payments. This coincides with the typical assumption about service providers. It, however, may for various purposes be interested in users' data access patterns, e.g., who are interested in what kinds of data at what locations and time. Although legislative approaches (say, opt-in or opt-out) can be adopted to regulate the collection of such information, it is much more assuring to prevent such privacy-intrusive behavior using sound technical means.

Network users are assumed to be *selfish*, *privacy-sensitive*, *curious*, and *rational*. By selfish, we mean that users always try to pay less for more data given any possible opportunity. For example, they may collude, use forged payments, or even compromise some entities responsible for access control. By privacy-sensitive and curious, we mean that users are reluctant to disclose their own data access patterns but are interested in learning others'. Users are also rational, meaning that they would misbehave only when benefiting from doing so. For instance, we assume that users do not launch DoS attacks on the sensor network because this is against their interest in acquiring useful sensed data. As another example, users do not attempt to evade access control by directly compromising many sensor nodes to read their data, which may require tremendous effort. Instead, users may only compromise a few sensor nodes if doing so could help them reuse tokens.

III. DP^{2}AC: Distributed Privacy-Preserving Access Control

In this section, we outline the DP^{2}AC scheme and defer the details of token-reuse detection (TRD) to Section IV. DP^{2}AC involves three phases: the *initialization* phase where the network owner picks security parameters, the *withdrawal* phase where users purchase tokens, and the *spending* phase where users spend tokens for data access.

A. System Initialization

DP^{2}AC is based on Chaum's blind signature protocol [20] which itself depends on RSA. The network owner creates its RSA public and private keys as $\langle n, e\rangle$ and d, respectively. Here, n is the product of two distinct random primes p and q; e, $1 < e < \phi$, is coprime to $\phi = (p-1)(q-1)$; d, $1 < d < \phi$, satisfies $ed = 1 \mod \phi$. The modulus n is typically at least 1024 bits long for sufficient security. In DP^{2}AC, the public key $\langle n, e\rangle$ is only used for verifying the network owner's signatures, so we select e to be 3 for very efficient signature verifications.

The network owner publishes $\langle n, e\rangle$ while keeping $\langle p, q, d\rangle$ confidential to himself. In particular, each sensor node is preloaded with $\langle n, e\rangle$ prior to network deployment. The network owner later could use authenticated broadcast such as μTESLA [21] to update sensor nodes with a new public key whenever needed. In addition, we assume that each user can get an authentic copy of $\langle n, e\rangle$, e.g., from the network owner's website or a public-key certificate binding $\langle n, e\rangle$ to the network owner issued by a trusted third party.

B. Token Withdrawal

Sensor network users need pre-buy some tokens from the network owner. Each token in DP^{2}AC consists of a λ-bit random integer and the network owner's signature on it,

where λ is a system parameter partially determining DP^{2}AC's correctness which we will discuss in Section IV-A.

Tokens can be purchased in many ways. Consider as an example user Alice who can purchase a token from the network owner through the following procedures:

1) Alice picks a λ-bit random integer m, $0 \leq m \leq 2^\lambda - 1 \leq n-1$, as well as a random secret integer k satisfying $0 \leq k \leq n-1$ and $\gcd(n,k) = 1$.
2) Alice sends $m^* = mk^e \mod n$ along with her payment information to the network owner.
3) The network owner returns $\sigma_m^* = (m^*)^d \mod n$ to Alice after verifying her payment information.
4) Alice computes $\sigma_m = k^{-1}\sigma_m^* \mod n$, which is the network owner's RSA signature on m.
5) Alice records the pair $\langle m, \sigma_m \rangle$ as a token.

The network owner is trusted to return a correct σ_m^*. It is easy to see that σ_m is a valid RSA signature on m, as $\sigma_m = k^{-1}\sigma_m^* = k^{-1}m^d k^{ed} = k^{-1}m^d k = m^d \mod n$. Due to the blinding factor k, the network owner cannot derive m and σ_m from m^*. In other words, given $\langle m, \sigma_m \rangle$, the network owner cannot link it to Alice. Each token corresponds to a monetary value and can be used to purchase an appropriate amount of sensed data. It is also possible to enable multi-denomination tokens by letting the network manager use a different RSA public/private key pair for each kind of denomination. For ease of presentation, we focus on single-denomination tokens throughput this paper.

Although unable to precisely associate individual tokens with the identities of their holders, the network owner may still narrow down the holder of a particular token to the users who purchased tokens. This might be a concern if the number of token buyers is limited. To overcome this, users may depend on a trusted third party to purchase tokens, thus avoiding submitting payment information directly to the network owner. Alternatively, the network owner can produce *token cards*, each containing a token covered by a scratch-off panel, and sell them via third parties such as chain stores. Users interested in the sensed data can then purchase token cards using cash or other payment methods if the card seller can be trusted.

C. Token Spending

The token-spending process is pretty simple. Consider Alice again as an example. After purchasing tokens, Alice (or her agent) can enter the sensor network to acquire data from any sensor node, say node A. Upon receiving a token $\langle m, \sigma_m \rangle$, node A first checks $m \stackrel{?}{=} (\sigma_m)^e \mod n$, a standard RSA signature verification. The check should succeed for a genuine token because $(\sigma_m)^e = m^{de} = m \mod n$. If so, m_i runs the Token-Reuse Detection process to make sure that $\langle m, \sigma_m \rangle$ was not used before. Only when $\langle m, \sigma_m \rangle$ passes both tests does A provide an appropriate amount of requested data to Alice that is commensurate with the token value. Since A cannot link $\langle m, \sigma_m \rangle$ to Alice, it does not know who requested the data as long as Alice does not disclose her identity. Alice's data access privacy is thus well protected. Also note that a signature verification takes an average of 0.7 seconds on TelosB motes [22], which may be significantly shortened if the assembly-language optimizations in [23] are used. So this operation is quite affordable in resource-constrained sensor networks.

IV. TRD: Token-Reuse Detection

Every token $\langle m, \sigma_m \rangle$ is simply a pair of numbers and unconditionally untraceable. Malicious users thus may unlimitedly reuse their tokens without worrying about being caught. It is therefore essential for sensor nodes to check whether received tokens have been used before answering data queries. This process is referred to as *token-reuse detection* (TRD) hereafter, which is part of the token-spending phase and occurs right after a token passes the signature verification test. Section I has discussed the significant shortcomings of the centralized TRD approach which depends on the base station. In this section, we present a suite of DTRD schemes without involving the base station. These schemes share the essential idea that a sensor node checks with some other nodes (called *witness*es) whether a token was used, but they differ in how the witnesses are chosen and the TRD accuracy/overhead.

A. Definitions and Performance Metrics

DTRD may have *false positive*s. Consider token $\langle m, \sigma_m \rangle$ as an example. Since m is a λ-bit random number, it is possible that another user might have picked the same number and thus owned the same token. If that user spent $\langle m, \sigma_m \rangle$ before Alice, then node A may determine $\langle m, \sigma_m \rangle$ from Alice to be a reused one even if Alice uses it for the first time, thus leading to a false positive. Assuming that the network owner issued M tokens, let us derive the false-positive probability, namely, the probability that at least two tokens are the same. The probability that all the M tokens are different is given by

$$\begin{aligned} \bar{P}(M) &= 1 \cdot (1 - \tfrac{1}{2^\lambda}) \cdot (1 - \tfrac{2}{2^\lambda}) \cdots (1 - \tfrac{M-1}{2^\lambda}) \\ &= \textstyle\prod_{k=1}^{M-1} (1 - \frac{k}{2^\lambda}) \\ &\approx \textstyle\prod_{k=1}^{M-1} e^{-\frac{k}{2^\lambda}} \quad (\text{since } 1 - x \approx e^{-x}) \\ &= e^{-\frac{M(M-1)}{2^{\lambda+1}}}. \end{aligned}$$

Then the false-positive probability with M tokens is

$$P(M) = 1 - \bar{P}(M) = 1 - e^{-\frac{M(M-1)}{2^{\lambda+1}}}.$$

If λ is sufficiently long, $P(M)$ can be made negligible for the maximum number of tokens the network owner may issue. For example, if $\lambda = 80$ and $M = 10^8$, $P(M) \approx 4 \times 10^{-9}$. To further reduce the false-positive probability, the network owner may also periodically change his RSA public/private keys, in which case he need refund users for unspent tokens. In this paper, we assume that false positives are negligible.

DTRD may also have *false negatives*, which occur when a reused token is mistaken as an unused one. Zero false negatives are obviously desirable, but to achieve them may incur significant overhead. It may be more realistic to tolerate a few false negatives with reasonable overhead.

Without loss of generality, we consider the following illustrative example hereafter. Assume that user Alice has successfully spent token $\langle m, \sigma_m \rangle$ for $r-1$ times, $r \geq 1$. Now

Alice attempts the rth use of $\langle m, \sigma_m \rangle$ at a non-compromised node A with which she has not spent $\langle m, \sigma_m \rangle$. This is a token-reuse attempt for $r \geq 2$. Assuming that $\langle m, \sigma_m \rangle$ passes the signature verification, A needs to further check whether $\langle m, \sigma_m \rangle$ is a reused one. We accordingly define the following DTRD performance metrics.

- p_r**–TRD probability**: This is defined as the probability of the rth use of $\langle m, \sigma_m \rangle$ being detected as a reuse attempt given that its first $(r-1)$ uses are successful.
- $\mathcal{C}_r$**–communication cost**: Since TRD requests and responses are all short messages, we assume the same cost to transmit and receive a TRD request or response across each hop for simplicity. $\mathcal{C}_r$ is defined as the number of hop-wise message transmissions incurred by the rth TRD of token $\langle m, \sigma_m \rangle$.
- $\mathcal{S}_r$**–storage cost**: S_r is defined as the storage space in the unit of tokens that sensor nodes totally spent for token $\langle m, \sigma_m \rangle$ after its rth attempted use.

We also let N be the total number of sensor nodes, θ be the number of compromised nodes, R be the circular transmission range of each node, and $\bar{L}$ be the average number of hops between two random nodes. We also assume effective mechanisms to ensure reliable end-to-end packet transmissions between any two nodes.

B. Scheme 1: Network-wide Flooding

In this scheme, every node is its own token witness and records all the tokens that were used at itself or all the others. On receiving token $\langle m, \sigma_m \rangle$, node A first checks its local storage to see whether m is there. If so, A considers $\langle m, \sigma_m \rangle$ a reused one; otherwise, A considers $\langle m, \sigma_m \rangle$ a fresh one, records it, and then floods m to all the other nodes which will all record m in their local storage.

Security and performance analysis

THEOREM 1: Scheme 1 can detect the rth ($r \geq 2$) use of any token as a reuse attempt with probability $p_r = 1$, regardless of the number of compromised nodes.

Proof: The proof of this theorem is straightforward. After the first use of token $\langle m, \sigma_m \rangle$, all the nodes in the network have recorded m in their local storage and thus can detect any attempted reuse of it. Since every node makes the decision on its own, its TRD result is not affected by compromised nodes. Therefore, we have $p_r = 1, \forall r \geq 2$. □

In Scheme 1, the first TRD of $\langle m, \sigma_m \rangle$ incurs N message transmissions and results in every node storing m, while all subsequent TRDs can be done locally and cause no additional communication and storage costs. To facilitate the comparison with later schemes, we amortize the N message transmissions over subsequent TRDs. Therefore, we have $\mathcal{C}_r = N/r$ and $\mathcal{S}_r = N, \forall r \geq 1$, which might be significant in large-scale sensor networks with very large N.

C. Scheme 2: Randomized Mapping

In this scheme, on receiving $\langle m, \sigma_m \rangle$, node A selects β witnesses $\{w_i\}_{i=1}^{\beta}$ of token $\langle m, \sigma_m \rangle$ as the nodes closest to locations $\mathcal{F}(m, s) = \{l_i\}_{i=1}^{\beta}$, where $\mathcal{F}$ denotes a good hash function and s is a random number. Then A sends a TRD request containing m to each witness using a geographic routing scheme such as GPSR [24] and sets a timer to the estimated longest message round-trip time. When receiving the TRD request, each witness w_i records m in its local storage if m is not found there; otherwise, w_i returns a TR alarm to node A. If node A receives any TR alarm before its timer expires, it considers token $\langle m, \sigma_m \rangle$ a reused one and a fresh one otherwise.

Security and performance analysis

THEOREM 2: Scheme 2 can detect the rth ($r \geq 2$) use of any token as a reuse attempt with probability $p_r \approx \frac{\beta^2(r-1)(N-\theta)}{N^2}$.

Proof: Each use of token $\langle m, \sigma_m \rangle$ generates β witnesses. Let s_i denote the random number used in generating witness locations for the ith use. If a node is the closest to multiple witness locations in $\{\mathcal{F}(m, s_i)\}_{i=1}^{r-1}$, it will be selected as a witness multiple times. Assuming that independently generated $\{s_i\}_{i=1}^{r-1}$ are mutually different, we now estimate $N_w(r-1)$, the average number of distinct witnesses after token $\langle m, \sigma_m \rangle$ has been successfully used $r-1$ times (including $\min\{0, r-2\}$ reuses). The probability that a node is not a witness in any use is given by $(1-\beta/N)^{r-1}$. Then the probability that a node is a witness in at least one use is $1-(1-\beta/N)^{r-1}$. We thus have $N_w(r-1) = N(1-(1-\beta/N)^{r-1})$. If $\beta/N \ll 1$, $N_w(r-1) \approx N(1-(1-\beta(r-1)/N)) = \beta(r-1)$ since $(1+x)^{\delta} \approx 1+\delta x$. Therefore, overlapping witnesses have no much impact on the performance of Scheme 3.

Now we estimate p_r. Since Alice does not know which nodes are witnesses, she can only randomly compromise some nodes among the total N nodes. Assuming that Alice has compromised $\theta < N$ nodes, the expected number of compromised witnesses is $\theta(1-(1-\beta/N)^{r-1}) \approx \lfloor (r-1)\beta\theta/N \rfloor$. There are thus $\lceil \beta(r-1)(1-\theta/N) \rceil$ non-compromised witnesses left. If none of them is selected as one of the β new witnesses, the rth TRD fails, which occurs with probability $(1-\beta/N)^{\lceil \beta(r-1)(1-\theta/N) \rceil}$. We thus have

$$p_r = 1-(1-\beta/N)^{\lceil (r-1)\beta(1-\theta/N) \rceil} \approx \frac{\beta^2(r-1)(N-\theta)}{N^2}. \quad (1)$$

□

In Scheme 2, the rth TRD results in $\mathcal{C}_r = (\beta + W_r)\bar{L}$ message transmissions, where W_r denotes the number of non-compromised witnesses that send a TR alarm to node A. If none of the $(r-1)\beta$ witnesses is compromised, each will return a TR alarm with probability β/N, resulting in totally $W_r = (r-1)\beta^2/N$ TR alarms on average. We thus have $\mathcal{C}_r = (1+(r-1)\beta/N)\beta\bar{L}, \forall r \geq 1$. In addition, the storage cost of Scheme 2 is $\mathcal{S}_r \approx r\beta, \forall r \geq 1$.

In contrast to Scheme 1, Scheme 2 has much lower communication and storage costs at the sacrifice in TRD capability. For example, if $N = 10,000$, $\beta = 20$, and $\theta = 10$, Scheme 2 can detect the first reuse of token $\langle m, \sigma_m \rangle$ with probability $p_2 = 0.04$ and the second reuse with probability $p_3 = 0.08$.

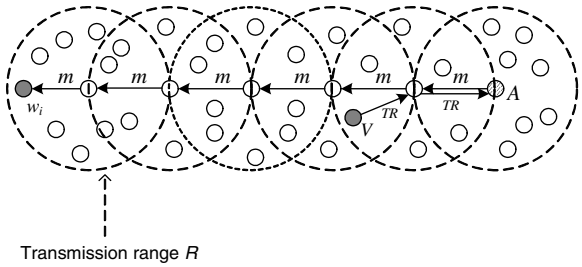


Fig. 1. Illustration of Scheme 3.

Since such results are certainly unsatisfactory, we propose another randomized DTRD scheme.

D. Scheme 3: Randomized Mapping Plus

Scheme 3 can greatly improve the TRD capability of Scheme 2 without additional storage cost. The idea is pretty simple. Due to the broadcast nature of radio transmissions, the TRD request for $\langle m, \sigma_m \rangle$ can be overheard by all the nodes within the transmission range R of its forwarding path. Scheme 3 allows every such node to return a TR alarm to A if it stores m. In this way, we can significantly improve the TRD probability for the same β.

Consider Fig. 1 as an example. Node w_i is one of the β witnesses selected by node A in the rth TRD for $\langle m, \sigma_m \rangle$ and was not chosen as a witness for $\langle m, \sigma_m \rangle$ in its past $r-1$ (re)uses. w_i will thus record m and return no TR alarm to A. Node V, however, acted as a witness for $\langle m, \sigma_m \rangle$ and can overhear the TRD request destined for w_i. Unlike Scheme 2, Scheme 3 permits V to return a TR alarm to node A.

Security and performance analysis

THEOREM 3: Assuming that the N sensor nodes are uniformly distributed over a field of area S, Scheme 3 can detect the rth ($r \geq 2$) use of any token as a reuse attempt with probability $p_r \approx 1 - (1 - S_\beta/S)^{\lceil \beta(r-1)(1-\theta/N) \rceil}$, where

$$S_\beta \approx \frac{((2\bar{L}\beta - 4\beta + 6)\pi + 3\sqrt{3}(\bar{L} - 1)\beta)R^2}{6}$$

is the area within which each node overhears at least one of the β TRD requests.

Proof: We first compute S_β. Recall that $\bar{L}$ denotes the average number of hops between any two nodes in the network. For simplicity, we assume that each forwarding path of a TRD request consists of exactly $\bar{L}+1$ nodes from node A which are separated by the transmission range R (cf. Fig. 1). The area of a transmission circle is $S_T = \pi R^2$, and the intersection area of two adjacent transmission circles can be easily calculated as $S_I = \frac{(4\pi - 3\sqrt{3})R^2}{6}$. Then the total area formed by the $\bar{L}$ transmission circles is given by

$$S_{\bar{L}} = \bar{L}S_T - (\bar{L} - 1)S_I = \frac{(2(\bar{L} + 2)\pi + 3\sqrt{3}(\bar{L} - 1))R^2}{6}.$$

Since there are β TRD requests, we have

$$\begin{aligned} S_\beta &\approx \beta S_{\bar{L}} - (\beta - 1)S_T \\ &= \pi R^2 + \frac{(2(\bar{L} - 1)\pi + 3\sqrt{3}(\bar{L} - 1))\beta R^2}{6}, \end{aligned} \tag{2}$$

where we assume that the radio coverage of different TRD requests do not overlap except around node A.

Now we estimate p_r. Since Alice does not know which nodes are witnesses, she can only randomly compromise some nodes among the total N nodes. Assuming that Alice has compromised $\theta < N$ nodes, the expected number of compromised witnesses is $\theta(1 - (1 - \beta/N)^{r-1}) \approx \lfloor (r-1)\beta\theta/N \rfloor$. There are thus $\lceil \beta(r-1)(1-\theta/N) \rceil$ non-compromised witnesses left. If none of them overhears or receives a TRD request, the rth TRD of token $\langle m, \sigma_m \rangle$ will fail, which occurs with probability $(1 - S_\beta/S)^{\lceil \beta(r-1)(1-\theta/N) \rceil}$. We thus have

$$p_r = 1 - (1 - S_\beta/S)^{\lceil \beta(r-1)(1-\theta/N) \rceil}. \tag{3}$$

□

Similar to Scheme 2, Scheme 3 has a communication cost of $\mathcal{C}_r = (\beta + W_r)\bar{L}, \forall r \geq 1$. Assuming that none of the $(r-1)\beta$ witnesses are compromised, each will return a TR alarm with probability S_β/S. We thus have $W_r = (r-1)S_\beta\beta/S$ and $\mathcal{C}_r = (1 + (r-1)S_\beta/S)\beta\bar{L}$. In addition, the storage cost of Scheme 3 is $\mathcal{S}_r \approx r\beta$, the same as Scheme 2 for the same β. In contrast to Scheme 2, Scheme 3 has much better TRD capability with a much smaller β as well as much smaller communication and storage costs, which we will see more clearly in later numerical and simulation results.

E. Scheme 4: Double Ruling

For both Scheme 2 and Scheme 3, there is a tradeoff between the TRD probability and the communication and storage costs: the larger β, the higher p_r, the larger $\mathcal{C}_r$ and $\mathcal{S}_r$, and versa. Now we introduce another scheme without this limitation. Scheme 4 is motivated by the double-ruling (DR) techniques [25] for data dissemination and query in sensor networks. The DR techniques aim at storing the sensed data along a continuous curve, called *replication curve*, instead of one or multiple isolated sensor nodes. Later users can query the data along another continuous curve, called *query curve*. As long as two curves intersect, users can retrieve the data of interest. Due to the space of limitation, we only introduce how to build Scheme 4 on the simplest DR technique: rectilinear DR. The extension of Scheme 4 to use other DR techniques in [25] is part of our future work. In the rectilinear DR technique, replication curves follow horizontal lines, while query curves follow vertical lines. If the sensor field has a regular topology (e.g., a square or rectangular), every replication curve will intersect with every query curve.

In Scheme 4, each token is treated as a unique data type as well as the information to be replicated and queried. Upon receiving a token for accessing data, each node sends a TRD request along a randomly positioned vertical line spanning the sensor field. If the TRD request hits any non-compromised node (witness) that records the token, that node will return a TR alarm to the TRD initiator. Otherwise, the token is considered fresh, and the TRD initiator replicates the token along a randomly positioned horizontal line on which each node should record the token.

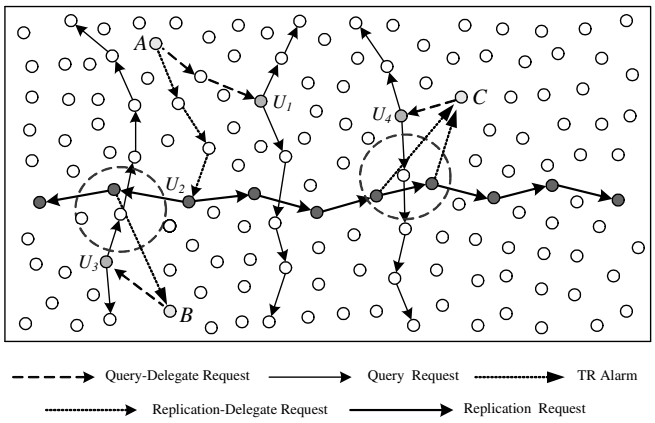


Fig. 2. Illustration of Scheme 4.

Specifically, on receiving token $\langle m, \sigma_m \rangle$, node A generates a random location $\mathcal{F}(m, s_1)$, where s_1 is a random number, and then sends a query-delegate request containing m to $\mathcal{F}(m, s_1)$ using GPSR [24]. The closest node to location $\mathcal{F}(m, s_1)$, denoted by U_1, finally receives the query-delegate request and is called a *query delegate* of node A. If U_1 finds m in its local storage, it returns a TR alarm to A; otherwise it sends two TRD requests containing m along vertical lines to the upper and lower network boundaries using GPSR [24], respectively. Since sensor nodes are randomly deployed and GPSR uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination, the actual routing paths followed by the TRD requests are more likely irregular curves resembling vertical query lines. In addition, there might not be a node at the exact intersection of the query line with an existing replication line (if any) for token $\langle m, \sigma_m \rangle$. Therefore, each node either receiving or overhearing the TRD request should return a TRD alarm to node A, and we call all such nodes as *intersection nodes*. If node A receives any TRD alarm before its timer fires, it considers token $\langle m, \sigma_m \rangle$ a reused one. Otherwise, it considers token $\langle m, \sigma_m \rangle$ a fresh one and then generates a random location $\mathcal{F}(m, s_2)$, where s_2 is a random number different from s_1. Finally, A sends a replication-delegate request containing m to $\mathcal{F}(m, s_2)$ using GPSR [24]. The closest node to location $\mathcal{F}(m, s_2)$, denoted by U_2, finally receives the query-delegate request and is called a *replication delegate* of node A. Subsequently, U_2 stores m into its local storage and also sends two replication requests containing m along horizontal lines to the left and right network boundaries using GPSR [24], respectively. Each node receiving the replication request should record m into its local storage.

To shed more light on Scheme 4, consider Fig. 2 as an example. Since the TRD requests from A's query delegate U_1 do not hit any replication curve in both up and right directions, A cannot receive any TRD alarm and thus generates a replication curve through its replication delegate U_2. Subsequently, suppose that Alice attempts reusing token $\langle m, \sigma_m \rangle$ at node B which selects U_3 as its query delegate. The TRD request from U_3 in the up direction can be overheard by node W_1 on the replication curve passing U_2 which will send a TR alarm back to B if not compromised. In some cases, there can be multiple intersection nodes in the replication curve overhearing the same TRD request. For example, suppose that Alice later tries reusing token $\langle m, \sigma_m \rangle$ at node C which chooses node U_4 as a query delegate. The TRD request from U_4 in the down direction can be overheard by two nodes W_2 and W_3 on the replication curve passing U_2. If either is not compromised and thus sends a TR alarm to C, the TR attempt can be thwarted. Since query and replication delegates are randomly chosen, Alice can only randomly compromise sensor nodes beforehand in the hope that some will happen to be the intersection nodes.

Security and performance analysis

After Alice has successfully used token $\langle m, \sigma_m \rangle$ for $r-1$ times, there will be $r-1$ horizontal replication curves in the network. We have the following theorem for p_r.

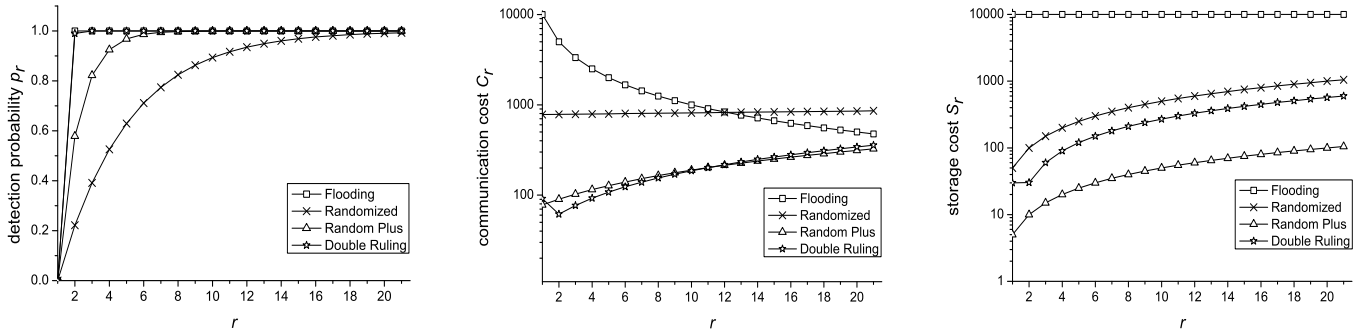
THEOREM 4: Scheme 4 can detect the rth use ($r \geq 2$) of any token as a reuse attempt with probability

$$p_r \approx 1 - (\frac{\theta}{N})^{r-1}. \quad (4)$$

Proof: After $r-1$ successful uses of token $\langle m, \sigma_m \rangle$, there are $r-1$ randomly positioned replication curves, along each of which at least one node can overhear the TRD request. For simplicity, we assume that only one node can overhear the TRD request, so there will be totally $r-1$ intersection nodes. Since the query curve chosen by node A is unpredictable, Alice can only randomly compromise some nodes in the network in order to defeat the rth TRD by A. Assuming that Alice has compromised θ nodes, then each of the $r-1$ intersection nodes is compromised with probability θ/N. The rth TRD will fail if all these intersection nodes are compromised, and the probability of this occurring is $(\frac{\theta}{N})^{r-1}$. We thus have $p_r \approx 1 - (\frac{\theta}{N})^{r-1}$. □

It is possible that Alice knows some of the $r-1$ replication curves, e.g., after compromising some nodes on them. To maximize her chance of escaping the TRD, she may divide the additional number θ of node she intends to further compromise into two portions instead of continuously compromising random sensor nodes. One portion is randomly selected from the known replication curves, and the other is randomly selected from those not on the known replication curves. For lack of space, we leave the investigation on the user's optimal attack strategy and related defenses to the future work.

For the communication cost, there are two possible cases: the rth token use either succeeds or not. In the first case, the communication cost consists of two parts: one incurred by the query, denoted by $\mathcal{C}_{r,1}$, and the other incurred by the replication, denoted by $\mathcal{C}_{r,2}$. Note that $\mathcal{C}_{r,1}$ includes the costs associated with transmitting the query-delegate request and two TRD requests, while $\mathcal{C}_{r,2}$ includes the costs associated with transmitting the replication-delegate request and two replication requests. Assuming that the sensor field is a rectangle with length D_L and width D_W, a horizontal line and a vertical line approximately amounts to $\lceil D_L/R \rceil$ and $\lceil D_W/R \rceil$ hops, respectively. Recall that $\bar{L}$ is the average number of



(a) detection probability p_r (b) communication cost $\mathcal{C}_r$ (c) storage cost $\mathcal{S}_r$

Fig. 3. Numeric result: p_r/$\mathcal{C}_r$/$\mathcal{S}_r$ vs. β.

hops between any two nodes. We have $\mathcal{C}_r = \mathcal{C}_{r,1} + \mathcal{C}_{r,2} = (\bar{L}+\lceil D_W/R\rceil)+(\bar{L}+\lceil D_L/R\rceil) = 2\bar{L}+\lceil D_W/R\rceil+\lceil D_L/R\rceil$. If the rth use fails, $\mathcal{C}_r$ is the sum of $\mathcal{C}_{r,1}$ and the cost incurred by TRD alarms. If none of the $r-1$ intersection nodes are compromised, there will be $r-1$ TR alarms sent back to A. It follows that $\mathcal{C}_r = \bar{L}+\lceil D_W/R\rceil+(r-1)\bar{L} = r\bar{L}+\lceil D_W/R\rceil$. Integrating these two cases, we have

$$\begin{aligned}\mathcal{C}_r =&(1-p_r)(2\bar{L}+\lceil D_L/R\rceil+\lceil D_W/R\rceil)\\ &+p_r(r\bar{L}+\lceil D_W/R\rceil)\\ =&(2+rp_r-2p_r)\bar{L}+(1-p_r)\lceil D_L/R\rceil+\lceil D_W/R\rceil.\end{aligned} \tag{5}$$

In addition, Scheme 4 has a storage cost

$$\mathcal{S}_r \approx (r-1)\lceil D_L/R\rceil+(1-p_r)\lceil D_L/R\rceil = (r-p_r)\lceil D_L/R\rceil. \tag{6}$$

Note that both $\mathcal{C}_r$ and $\mathcal{S}_r$ hold for all $r \geq 1$ since $p_1 = 0$.

F. Discouraging token-reuse attempts

In Schemes 2∼4 with random witness selection, the TRD probability will increase with the number of successful token reuses. Since user identities are not revealed, however, a user may keep trying to reuse tokens at different nodes even if the probability of a successful token reuse becomes vanishingly small. The resulting TRD processes will cause significant communication and storage overhead. It is thus necessary to put forward some countermeasures to deter such misbehavior. One possible solution is to let a sensor node provide arbitrarily wrong data to the user when detecting his token-reuse attempt. If the user cannot differentiate true data from fake data, further processing the data will cause undesirably severe consequences especially when the data are used as the basis for critical decision making. Assuming that users are selfish yet rational and know our DTRD scheme in use, they will be less motivated to attempt reusing tokens if knowing that their attempts can be detected with a sufficiently high probability.

V. Performance Evaluation

In this section, we evaluate the performance of our DTRD schemes using both numeric and simulation results.

A. Numeric Results

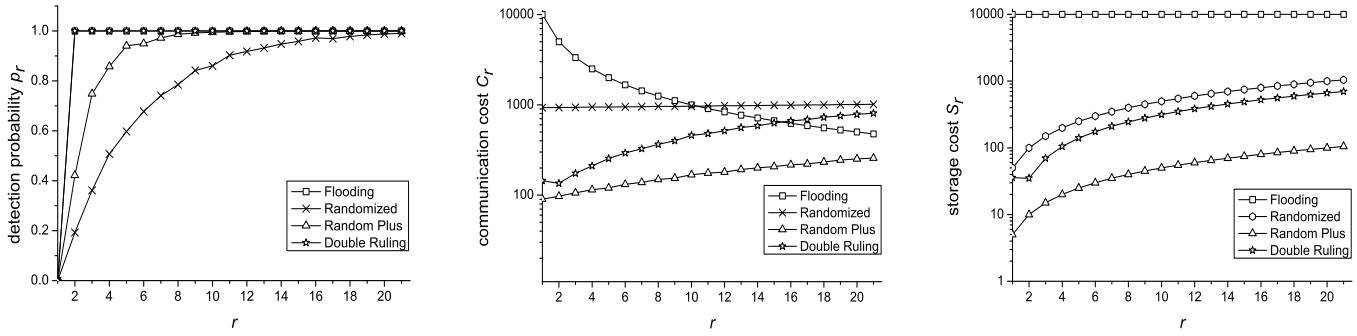
We assume a square sensor field of 3000×3000 units within which 10000 nodes are uniformly distributed. The transmission range of each node is 100 units, which ensures an almost fully connected network [26]. The average number of hops between any two nodes is thus $\bar{L} \approx 15$, which can be easily calculated according to [27]. In addition, the numbers of chosen witnesses (i.e., β) per TRD in Scheme 2 and Scheme 3 are set to 50 and 5, respectively. Different values of β are used here due to their inherently different TRD capabilities. We also temporarily assume that there are 100 compromised nodes and will simulate the impact of different numbers of compromised nodes in Section V-B.

Fig. 3(a) compares the TRD probabilities of the four DTRD schemes, which vary with the number of token uses. Note that a token-reuse (TR) attempt occurs for $r \geq 2$. As we can see, Scheme 1 (flooding) can always detect any TR attempt, and Scheme 4 (double ruling) whose TRD probability increases with r can detect any TR attempt almost for sure. In contrast, Scheme 2 (randomized) has poor TRD capability due to its completely randomized witness selection, while Scheme 3 (randomized plus) can almost detect the reuse attempt of a token after it was successfully used a few times (e.g., 8 times).

Fig. 3(b) and Fig. 3(c) compare their communication and storage costs in $\log 10$ scale, respectively. We can see that Scheme 1 has much larger communication and storage costs than Scheme 3 and Scheme 4 for similar TRD probabilities, which make it less suitable for large-scale sensor networks. In addition, Scheme 4 outperforms Scheme 3 in the communication cost for small r values (e.g., $r \leq 8$), but it is worse in the storage cost.

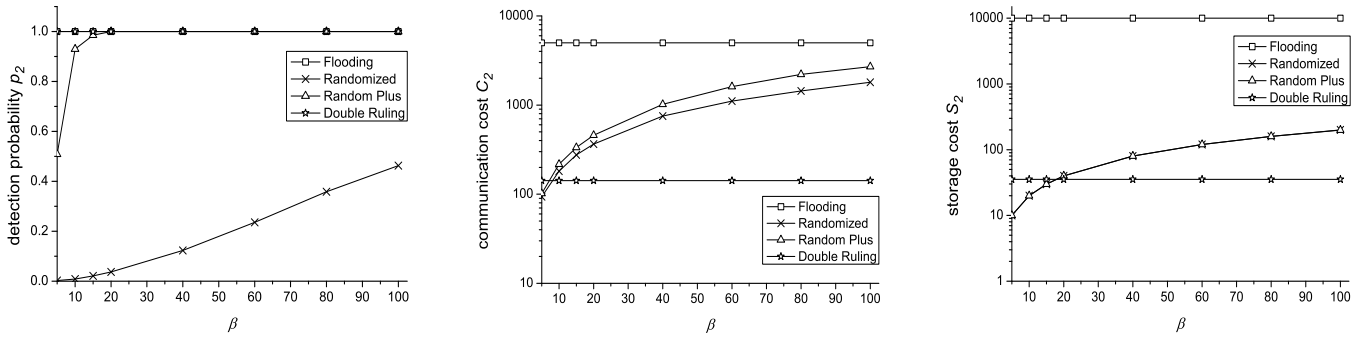
B. Simulation Results

We also did simulations using the network configurations in Section V-A, unless otherwise stated. For our purpose, the simulation code was written in C++, and we assume error-free and collision-free packet transmissions. Each point in Figs. 4∼6 represents the average of the results of spending a random token at 100 random nodes. For each token spending, we also simulated 100 different random sets of compromised



(a) detection probability p_r (b) communication cost $\mathcal{C}_r$ (c) storage cost $\mathcal{S}_r$

Fig. 4. Simulation result: p_r/$\mathcal{C}_r$/$\mathcal{S}_r$ vs. r.



(a) detection probability p_2 (b) communication cost $\mathcal{C}_2$ (c) storage cost $\mathcal{S}_2$

Fig. 5. Simulation result: p_r/$\mathcal{C}_r$/$\mathcal{S}_r$ vs. β.

nodes and took the average. In addition, the 95% confidence intervals for p_r and $\mathcal{C}_r$ are both less than 10% of the sample mean. Moreover, we simulated many different random tokens and network topologies. Since the results are quite consistent with what we report here, we ignore them for lack of space.

Fig. 4 shows the simulation results corresponding to the numerical results in Fig. 3. We assumed in the theoretical analysis that witnesses are mutually different, but there are actually overlapping witnesses in the simulations. Since the number of non-compromised witnesses in simulations is smaller than that in numerical calculations, the numerical TRD probabilities are slightly larger than the corresponding simulated ones. In addition, there may be multiple nodes on a replication curve overhearing and responding to a TRD request (cf. Section IV-E) in the simulations in contrast to the single node assumed in the theoretical analysis. It is not surprising to observe that the communication cost of Scheme 4 is larger than that of Scheme 3. In general, however, the simulation results are quite consistent with the numerical results.

Fig. 5 demonstrates the impact of β, the number of selected witnesses per TRD, on Scheme 2 and Scheme 3, where $r = 2$ because we are more concerned about detecting the user's first TR attempt. Note that Scheme 1 and Scheme 4 are not affected by β, and they are shown here just for the comparison purpose. Generally speaking, the larger β, the higher the TRD probabilities, the larger the communication and storage costs, and vice versa. This coincides with the intuition. In addition, $\beta = 10$ is sufficient for Scheme 3 to detect the first TR attempt with probability $p_2 = 0.9$, while Scheme 2 can only achieve $p_2 < 0.5$ even with $\beta = 100$. The communication cost of Scheme 3, however, grows faster than that of Scheme 2. The reason is that Scheme 3 allows not only the chosen witnesses but also other nodes that record the corresponding token and overhear the TRD request to return a TR alarm, leading to more TR alarms than in Scheme 2. Furthermore, Scheme 3 has comparable TRD capability to that of Scheme 4 with $\beta = 20$, in which case it has larger communication and storage costs.

Fig. 6 shows the impact of the number θ of compromised nodes on the TRD capabilities of the four schemes, where $r = 2$, $\beta = 100$ for Scheme 2, and $\beta = 10$ for Scheme 3. We can see that Scheme 1 is not affected by θ because it is a deterministic scheme in which each node is its own token witness. The other three schemes are also insensitive to θ due to randomized witness selection. For example, even when 20 percent of the sensor nodes are compromised, Scheme 4 can still detect the first TR attempt with probability $p_2 = 0.97$, which is higher than the numerical result 0.8. The reason is that there might be multiple nodes on a replication curve reply-

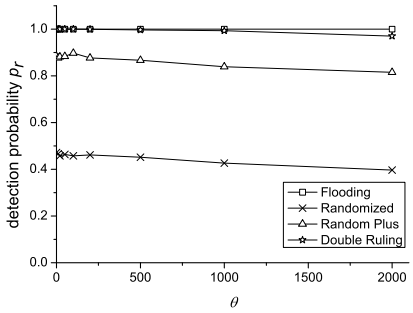


Fig. 6. Detection probability p_r vs. θ.

ing to the same TRD request rather than just one node assumed in the theoretical analysis. These results show that our DTRD schemes have strong resilience against node compromise.

C. Discussion

We summarize the performance evaluation as follows.

- Scheme 1 can detect any TR attempt with probability 1 with prohibitive communication and storage costs.
- Scheme 2 has very poor TRD capability or incurs significant communication and storage costs to achieve a satisfactory TRD probability.
- Scheme 3 has very good TRD capability with reasonable communication and storage costs increasing with the number of selected witnesses per TRD and the number of successful token reuses.
- Scheme 4 has very good TRD capability with reasonable communication and storage costs increasing with the field size and the number of successful token reuses.

Since users are assumed to be rational, they would have no incentives to reuse tokens with Scheme 1 and Scheme 4 in place and may only attempt to reuse a token for a few times (say, 6) with Scheme 3 in place, especially when the discouraging measure in Section IV-F is deployed. In practice, Scheme 4 may be the best choice for large-scale sensor networks, followed by Scheme 3. The numerical and simulation results confirm that DP^{2}AC is a very practical and trustworthy solution for sensor networks.

VI. Conclusion

In this paper, we presented DP^{2}AC, a novel token-based approach to achieve distributed privacy-preserving access control in single-owner multi-user sensor networks. The efficacy and efficiency of DP^{2}AC are confirmed by detailed performance evaluations. As the future work, we intend to investigate more efficient DTRD techniques for DP^{2}AC under different attacker models, e.g., adaptive attackers.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants CNS-0716302 and CNS-0831963. The authors also would like to thank Dr. Jie Gao and Mr. Rik Sarkar for sharing the simulation code of the Double Rulings scheme.

References

[1] B. Sheng, Q. Li, and W. Mao, "Data storage placement in sensor networks," in *ACM MobiHoc'06*, Florence, Italy, May 2006, pp. 344–355.
[2] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," in *ACM MobiHoc'05*, Urbana-Champaign, IL, USA, May 2005, pp. 378–389.
[3] H. Wang and Q. Li, "Distributed user access control in sensor networks," in *DCOSS'06*, San Francisco, CA, June 2006, pp. 305–320.
[4] D. Liu, "Efficient and distributed access control in sensor networks," in *DCOSS '07*, Santa Fe, New Mexico, USA, June 2007.
[5] K. Ren, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," in *IEEE SECON'07*, San Diego, CA, June 2007, pp. 223–232.
[6] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pDCS: Security and privacy support for data-centric sensor networks," in *IEEE INFOCOM'07*, Anchorage, Alaska, USA, May 2007, pp. 1298–1306.
[7] Y. Zhou, Y. Zhang, and Y. Fang, "Access control in wireless sensor networks," *Ad Hoc Networks, Special Issue on Security in Ad Hoc and Sensor Networks*, vol. 5, no. 1, pp. 3–13, Jan. 2007.
[8] ORION, http://www.joiscience.org/ocean_observing/advisors.
[9] NOPP, http://www.nopp.org/.
[10] IOOS, http://www.ocean.us/.
[11] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *SECON '07*, San Diego, CA, USA, June 2007, pp. 203–212.
[12] K. Ren, W. Lou, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environment," in *IEEE Transactions on Vehicular Technology*, 2006.
[13] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *ACM CCS'03*, Washington, DC, Oct. 2003, pp. 62–72.
[14] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *S&P'03*, Oakland, CA, May 2003, pp. 197–213.
[15] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *ACM CCS'03*, Washington, DC, Oct. 2003, pp. 52–61.
[16] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Select. Areas Commun., Special Issue on Security in Wireless Ad Hoc Networks*, vol. 24, no. 2, pp. 247–260, Feb. 2006.
[17] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology - Crypto '82*. Springer-Verlag (1983), 1982, pp. 199–203.
[18] I. Osipkov, E. Y. Vasserman, N. Hopper, and Y. Kim, "Combating double-spending using cooperative P2P system," in *ICDCS'07*, Toronto, Canada, June 2007.
[19] J.-H. Hoepman, "Distributed double spending prevention," in *15th Int. Workshop on Security Protocols*, Brno, Czech Republic, Apr. 2007.
[20] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Comm. ACM*, vol. 28, no. 10, pp. 1030–1044, Oct. 1985.
[21] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *ACM Wireless Networks*, vol. 8, no. 5, pp. 521–234, Sep. 2002.
[22] P. Dutta, J. Hui, D. Chu, and D. Culler, "Securing the deluge network programming system," in *IPSN'06*, Nashville, Tennessee, Apr. 2006, pp. 326–333.
[23] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *CHES'04*, Boston, MA, Aug. 2004, pp. 119–132.
[24] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *ACM MOBICOM'00*, Boston, MA, Aug. 2000, pp. 243–254.
[25] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," in *ACM MOBICOM'06*, Los Angeles, California, USA, Sept. 2006, pp. 286–297.
[26] P. Gupta and P. R. Kumar, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*. Boston: Birkhauser, 1998, ch. Critical Power for Asymptotic Connectivity in Wireless Networks.
[27] L. E. Miller, "Distribution of link distances in a wireless network," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, pp. 401–412, 2001.