Accepted for publication in IEEE Transactions on Parallel and Distributed Systems

Distributed Privacy-Preserving Access Control in Sensor Networks

Rui Zhang, *Student Member, IEEE,* Yanchao Zhang, *Member, IEEE,* and Kui Ren, *Senior Member, IEEE*

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, for which we propose a suite of *distributed token reuse detection* (DTRD) schemes without involving the base station. These schemes share the essential idea that a sensor node checks with some other nodes (called *witnesses*) whether a token has been used, but they differ in how the witnesses are chosen. We 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.

Keywords—Wireless sensor networks, access control, privacy, security.

1 Introduction

The rapid advances in storage technology are making it economically practical to equip sensor nodes with high-capacity energy-efficient flash memories [2]. Such sensor nodes continuously monitor the physical world and record all kinds of data at local flash memories instead of sending the data in realtime to remote external storage. Sensor networks of this kind have numerous industrial and scientific applications, which need fine-granular long-term archival data to analyze historical trends, detect unusual patterns, and so on [2], [3].

Data stored at sensor nodes can be accessed in two ways. The first approach relies on a base station connecting the sensor network to the outside network, say, the Internet. Users interested in sensed data can issue data queries to sensor nodes through the base station which in turn forwards query results from sensor nodes to the users. If many queries are issued from lots of users, sensor nodes close to the base station have to always engage in relaying data to and from the base station and thus would quickly die out due to energy depletion. In addition, the base station may become the bottleneck and is the single point of failure. Furthermore, for sensor networks deployed in extreme and hazardous environments such as oceans and animal habitats, it may be impossible or prohibitive to maintain a stable communication connection between the base station and the outside network. The second approach allows users to freely roam in the sensor network and directly access sensed data without involving the base station. Same as [4], [5], [6], [7], [8], [9], [10], this paper is concerned with the latter approach.

Owners and users of sensor networks may be different, which necessitates privacy-preserving access control. For example, increasing programs and projects such as ORION [11], NOPP [12] and IOOS [13] 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 to users willing to pay. There is also a growing requirement for protecting users' data access privacy [3], [14]. In particular, a user may desire 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 [11], [12], [13] may want to hide its network regions of interest from both the network owner and other network users that might be potential business competitors [3].

Privacy-preserving access control in sensor networks has so far received little attention. Related work [5], [6], [7], [8], [9], [10] addresses access control by authenti-

The preliminary version of this paper appeared in IEEE INFOCOM'09 [1]
R. Zhang and Y. Zhang are with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 (email: {ruizhang, yczhang}@asu.edu).
K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 South Dearborn Street, Suite 103 Siegel Hall, Chicago, IL 60616. (email: kren@ece.iit.edu).

cating network users before granting them data access rights, but the privacy of users is not considered. As far as we know, SPYC [3] is the only work that takes into account the access privacy of network users. SPYC assumes the first data acquisition method mentioned above, i.e., 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 purchases some *token*s from the network owner before entering the sensor network, who can subsequently send a query with an unspent token to any sensor node. Once validating the token, the sensor node provides the user with an appropriate amount of requested data commensurate with the denomination of the token. Token generations involve blind signature [15], which leads 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 [15], [16], [17] 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 have financial interest in reusing tokens at different sensor nodes without worrying about being caught, which 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 do so, the base station need record every token submitted by sensor nodes. This centralized method has several limitations although it can detect every token-reuse attempt. First, the base station is the single point of failure: once compromising the base station, malicious users can freely reuse tokens. Second, if there are many tokens to verify, sensor nodes close to the base station would deplete their energy quickly for relaying TRD requests and replies. Third, 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, which makes DP^{2}AC a very practical and trustworthy solution for sensor networks..

The rest of this paper is organized as follows. Section 2 gives the network and trust models. Section 3 presents the DP^{2}AC scheme, followed by the TRD details in Section 4. We then introduce some extensions of the TRD schemes in Section 5 and evaluate the security and performance of DP^{2}AC in Section 6. Some additional related work is discussed in Section 7 and this paper is finally concluded in Section 8.

2 Network and Trust Models

2.1 Network Model

We assume a single-owner multi-user large-scale sensor network with N sensor nodes that 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 [11], [12], [13]. 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 [5], [6], [7], [8], [9], [10], we assume that sensor nodes know their geographical locations which can be acquired via many existing localization schemes such as [18], [19].

2.2 Trust Model

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 *privacy-sensitive*, *curious*, and *rational*. 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.

3 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 4. 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.

3.1 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. To enable efficient signature verifications, we select e to be $2^{16}+1$, which is commonly recommended in practce [21].

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 [22] 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 which is issued by a trusted third party.

3.2 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 4.1.

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, $0 \le m \le 2^\lambda - 1 \le n - 1$, as well as a random secret integer k satisfying $0 \le k \le n-1$ and $\gcd(n, k) = 1$.
2) Alice sends $m^* = h(m)k^e \mod n$ along with her payment information to the network owner, where $h(\cdot)$ is good one-way hash function of λ bits.
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 $h(m)$.
5) Alice records the pair $\langle m, \sigma_m\rangle$ as a token.

The network owner is trusted to return a correct σ^*_m. We can see that σ_m is a valid RSA signature on $h(m)$, as

$$\begin{aligned}
\sigma_m &= k^{-1}\sigma^*_m && \mod n && (1)\\
&= k^{-1}h(m)^d k^{ed} && \mod n && (2)\\
&= k^{-1}h(m)^d k && \mod n && (3)\\
&= h(m)^d && \mod n && (4)
\end{aligned}$$

Due to the blinding factor k, the network owner cannot derive $h(m)$ 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.

3.3 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 if $h(m) = (\sigma_m)^e \mod n$ holds, a standard RSA signature verification. The check should succeed for a genuine token because $(\sigma_m)^e = h(m)^{de} = h(m) \mod n$. If so, node A runs the Token-Reuse Detection process to make sure that $\langle m, \sigma_m\rangle$ has not been 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 0.79 seconds on MICAz motes [23]. So this operation is quite affordable in resource-constrained sensor networks.

4 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. Section 1 has discussed the significant shortcomings of the centralized TRD approach which depends on the base station. In this section, we present a suite of distributed TRD schemes without involving the base station.

4.1 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, which is difficult for the network owner to detect due to the blinding factor k used in token withdrawn process. 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 of all the M tokens being 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 $M = 10^8$, then $P(M) \approx 4 \times 10^{-9}$. 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 previous $r - 1$ uses are successful. We apparently have $p_1 = 0$ assuming negligible false positives.
- $\mathcal{C}_r$**–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**: $\mathcal{S}_r$ 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.

4.2 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. Specifically, 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

Assuming that a reliable network flooding scheme such as [24] is used, we then have the following theorem regarding the TRD capability of Scheme 1.

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.*

The proof is straightforward and thus omitted.

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 $\mathcal{S}_r = N$, for all $r \geq 1$, which might be significant in large-scale sensor networks with very large N.

4.3 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 GFG [25], [26], 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,

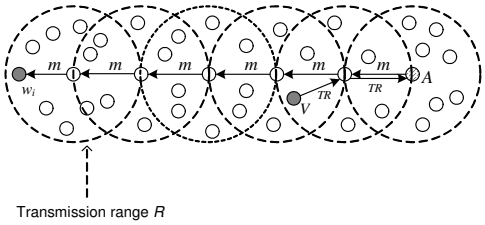


Fig. 1. Illustration of Scheme 3.

w_i returns a TR alarm to node A. If node A receives any TR alarm before its timer expires, it considers $\langle m, \sigma_m \rangle$ 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 $p_r \approx \frac{\beta^2(r-1)(N-\theta)}{N^2}$, where $\theta < N$ is the number of compromised nodes.*

We give the proof of the above theorem in the supplemental file.

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 total $W_r = (r-1)\beta^2/N$ TR alarms on average. We thus have $\mathcal{C}_r = (1 + (r-1)\beta/N)\beta\bar{L}$, for all $r \geq 1$. In addition, the storage cost of Scheme 2 is $\mathcal{S}_r \approx r\beta$, for all $r \geq 1$.

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 $p_3 = 0.08$. Since such results are certainly unsatisfactory, we propose another DTRD scheme.

4.4 Scheme 3: Randomized Mapping Plus

Scheme 3 can greatly improve the TRD capability of Scheme 2 without any 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 greatly 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.

We give the proof of the above theorem in the supplemental file.

Similar to Scheme 2, Scheme 3 has a communication cost of $\mathcal{C}_r = (\beta + W_r)\bar{L}$, for all $r \geq 1$. 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.

4.5 Scheme 4: Rectilinear 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 vice versa. Now we introduce another scheme without this limitation.

Scheme 4 is motivated by the double-ruling (DR) techniques [27] for data dissemination and query in sensor networks, which can be viewed a variant of the quorum based location update scheme proposed in [28] to enable scalable routing in ad hoc wireless 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. In what follows, we introduce a TRD scheme built upon the simplest DR scheme, rectilinear DR. 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 curve horizontally spanning the sensor field on which each node should record the token.

In the original rectilinear DR technique, both query and replication lines are straight lines and known to pass

the query and replication initiator. This, unfortunately, enables the user to easily fail the TRD process by just compromising the nodes at the intersections of the query line and previous replication lines beforehand. Scheme 4 resolves this issue through randomly positioned query and replication lines.

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 GFG [25], [26]. 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 GFG [25], [26], respectively. Since sensor nodes are randomly deployed and GFG [25], [26] 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 another random number. Finally, A sends a replication-delegate request containing m to $\mathcal{F}(m, s_2)$ using GFG. The closest node to location $\mathcal{F}(m, s_2)$, denoted by U_2, receives the query-delegate request and is called a *replication delegate* of A. Subsequently, U_2 stores m into its local storage and sends two replication requests containing m along horizontal lines to the left and right network boundaries using GFG [25], [26], respectively. Each node receiving the replication request should record m into its local storage.

Security and performance analysis

The following theorems are about the TRD capability and the communication and storage cost of Scheme 4, whose proofs are given in the supplemental file.

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

$$p_r \approx 1 - (\frac{\theta}{N})^{r-1}. \tag{5}$$

Theorem 5. *Assuming that the sensor field is a rectangle with length D_L and width D_W. The communication and storage cost incurred by Scheme 4 are given by*

$$\mathcal{C}_r = (2 + rp_r - 2p_r)\bar{L} + (1 - p_r)\lceil D_L/R \rceil + \lceil D_W/R \rceil \ , \tag{6}$$

and

$$\mathcal{S}_r \approx (r - p_r)\lceil D_L/R \rceil \ , \tag{7}$$

respectively, where p_r is given in Eq. (5).

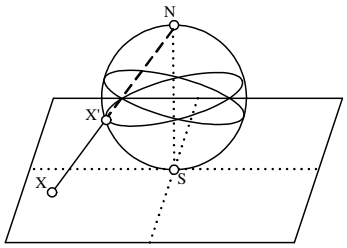


Fig. 2. Basic Idea of Scheme 5.

4.6 Scheme 5: Spherical Double Ruling

In this subsection, we introduce another scheme to significantly reduce the communication cost of Scheme 4 while providing similar overwhelming TRD probability.

Scheme 5 is a variation of the spherical double ruling scheme proposed in [27], which is based on stereographic mapping. For clarity, we use Fig. 2 to briefly introduce the basic idea of Scheme 5. Assuming that the physical shape of the sensor field is known and regular, we can place a virtual sphere with radius l tangent to the sensor field at the origin (the center of the field) which is also called the *south pole*. Each point X in the sensor field can be mapped to a unique point X' on the sphere, which is the intersection of the line through X and the *north pole* with the sphere. Conversely, each point on the sphere can be mapped to a point on the sensor field except the north pole if the sensor field is infinite. According to the property of stereographic mapping, every great circle on the sphere can be mapped to a circle in the sensor field, which we call *projected circle*. Intuitively, any two great circles on the sphere will intersect with each other at two points, so do any projected circles. Scheme 5 selects the replication/query curves as random projected circles, which are guaranteed to intersect with each other.

Let us first see some properties of projected circles. We have the following theorem regarding their locations and sizes, whose proof is given the supplemental file.

Theorem 6. *Any projected circle can be written in the following form*

$$(\mathbf{x} - ul)^2 + (\mathbf{y} - vl)^2 = (u^2 + v^2 + 4)l^2, \tag{8}$$

for some v and u.

We can see that any projected circle is a circle with center (ul, vl) and radius $l\sqrt{u^2 + v^2 + 4}$.

Intuitively, assuming that l is a system parameter chosen by the network owner, any projected circle can be determined by (u, v). Each node can thus select the replication/query curve by choosing u and v. There is, however, one more issue to be addressed. One may notice that since the sensor field is finite, if u and v are arbitrarily chosen, some part of a projected circle may be outside of the sensor field, so do the intersections of some projected circles. We thus must put some constraints on u and v to avoid such cases.

For simplicity, we assume that the sensor field is an $\mathbf{L}\times\mathbf{L}$ square. To ensure that a projected circle is within the sensor field, the following inequalities need be satisfied.

$$\begin{cases} ul + l\sqrt{u^2+v^2+4} \le \mathbf{L}/2, \\ ul - l\sqrt{u^2+v^2+4} \ge -\mathbf{L}/2, \\ vl + l\sqrt{u^2+v^2+4} \le \mathbf{L}/2, \\ vl - l\sqrt{u^2+v^2+4} \ge -\mathbf{L}/2. \end{cases} \tag{9}$$

Lemma 1. *Let $\mathbf{L} = \nu l$. If $\nu \ge 4$ and $u^2 + v^2 \le \delta^2$, where $\delta = \nu/4 - 4/\nu$, then the inequality in (9) hold.*

The proof of Lemma 1 is given in the supplemental file. According to Lemma 1, if $\mathbf{L} = 8l$, for example, then $u^2 + v^2 \le 9/4$ can ensure the inequalities hold. Note that Lemma 1 does not give the exact solution for the inequalities, which is not needed for our purpose. In what follows, we introduce the operation of Scheme 5.

In scheme 5, each sensor node is loaded with l and ν. These parameters can either be preloaded by the network owner before the deployment or broadcasted after the deployment.

Upon receiving $\langle m, \sigma_m\rangle$, the sensor node, say A, selects a point (u, v) from the disc $\{(u, v)|u^2+v^2 \le \delta^2\}$ uniformly at random and sends a TRD request consisting of m, u and v to the closest node on the projected circle (determined by u and v as in Eq. (8)) using GFG [25], [26]. The closest node on the circle, say C, which we call *TRD coordinator*, checks whether m is in its local memory. If so, it returns a TR alarm to node A. Otherwise, C inserts its node ID into the TRD request and forwards it counterclockwise along the projected circle using trajectory based forwarding [29] in a greedy fashion as in [27].

More specifically, node C sends the TRD request to the node which advances furthest along the circle in the required direction, say F_1. The TRD request will also be overheard by all the other one-hop neighbors of node C. All the nodes that overhear the TRD request search their local memories to see whether m is stored. If so, a TR alarm is returned to node C. Node F_1 also checks whether m is in its local memory and returns a TR alarm to node C if so. Otherwise, F_1 temporarily stores m and the ID of the node from which it receives the TRD request, i.e., C in this case.

One trick here is that node F_1 does not forward the TRD request to the next node immediately. Instead, it sets a timer T_1 to see whether any of C's other neighbors returns a TR alarm. In particular, assume that the processing time of a TRD request is τ. F_1 sets T_1 to be longer than τ plus twice of the message transmission time. The observation here is that if node C receives any TR alarm from its one-hop neighbors other than F_1, F_1 can overhear the TR alarm when C sends it back to node A. In such cases, it is unnecessary for node F_1 to further forward the TRD request; instead, F_1 stops T_1 and deletes m from its buffer. Only if F_1 does not overhear any TR alarm before T_1 fires, does it forward the TRD request to the next node on the circle and set another timer T_2 which is longer than the time for a TRD request to traverse the largest projected circle and a TR alarm to traverse backward. All the subsequent nodes on the circle will perform the same operation as F_1 does.

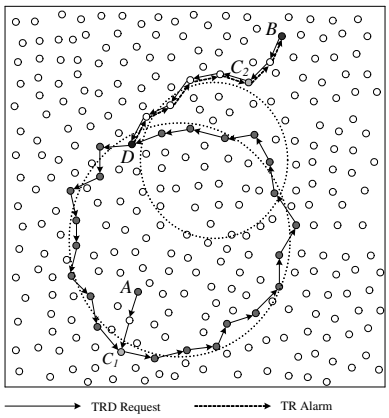


Fig. 3. Illustration of Scheme 5.

If any node returns a TR alarm, the alarm will be returned to node C along the circle in the clockwise direction which will further forwards it to the node A. In this way, those who temporarily store m can delete m from their memories since m is stored along at least one projected circles. If m has not been used before, the TRD request will eventually reach C. Note that the last node will forward the TRD request to C instead of in the greedy fashion. After the timer T_2 expires, each node along the projected circle stores m in its local memory. If A does not receive any TR alarm before the timer expires, it considers m valid and reused otherwise.

We give an example of Scheme 5 in Fig. 3. Assume that the token $\langle m, \sigma_m\rangle$ is used for the first time at node A. Node A chooses the projected circle on which node C_1 is the TRD coordinator. Node A then sends the TRD request to node C_1 using GFG which subsequently forwards it along the circle in the counterclockwise direction. Since none of the nodes in the network has previously stored m, the TRD request will traverse the projected circle and finally reach C_1. All the nodes along the circle that receive m will store m in their local memories after the their timers expire. Subsequently, suppose that Alice attempts reusing token $\langle m, \sigma_m\rangle$ at node B which selects another projected circle on which node C_2 is the TRD coordinator. The TRD request from C_2 can be received by node D on the previous projected circle and if D is not compromised, a TR alarm will be sent back to node C_2 and finally forwarded to node B. Therefore, the reuse attempt can be detected by node B.

Security and Performance Analysis

Now we analyze the performance of Scheme 5. Different from previous DTRD schemes, not all the nodes in the sensor field can be chosen as witness, e.g., the nodes at position $(0, 0)$ and $(\nu l/2, \nu l/2)$, so we first evaluate the *witness region*, which is the area where nodes are possible to become witnesses. Note that for Scheme 1∼4,

the witness regions are simply the whole network. We have the following theorem for the witness region of Scheme 5, whose proof is given in the supplemental file.

Theorem 7. *The witness region of Scheme 5 is a ring centered at* $(0,0)$ *with outer radius* $\mathbf{R_o} = \nu l/2$ *and inner radius* $\mathbf{R_i} = (\sqrt{\delta^2+4}-\delta)l$.

The following theorems are about the TRD capability and the communication and storage cost of Scheme 5, whose proofs are given in the supplemental file.

Theorem 8. *Scheme 5 can detect the* r*th,* $r \geq 1$ *reuse of any token with overwhelming probability*

$$p_r \geq 1 - (\frac{\theta \mathbf{L}^2}{\pi N(\mathbf{R_o}^2 - \mathbf{R_i}^2)})^{2r-2}, \tag{10}$$

where $\mathbf{R_o} = \nu l/2, \mathbf{R_i} = (\sqrt{\delta^2+4}-\delta)l$, *and* θ *is the number of compromised sensor nodes.*

Theorem 9. *The communication and storage cost incurred by Scheme 4 are given by*

$$\mathcal{C}_r = \frac{(1+p_r)E(\mathcal{L})}{R} + \frac{(2r-2-2rp_r+3p_r)\pi E(\mathcal{R})}{(r-1)R}\ , \tag{11}$$

for $r > 1$,

$$\mathcal{C}_1 = \frac{E(\mathcal{L}) + 2\pi E(\mathcal{R})}{R}\ , \tag{12}$$

for $r = 1$, *and*

$$\mathcal{S}_r \approx \frac{2(r-p_r)\pi E(\mathcal{R})}{R}, \tag{13}$$

respectively, where

$$E(\mathcal{L}) = \frac{1}{\pi \mathbf{L}^2 \delta^2} \iint\limits_{u^2+v^2 \leq \delta^2} \int_{-\frac{\mathbf{L}}{2}}^{\frac{\mathbf{L}}{2}} \int_{-\frac{\mathbf{L}}{2}}^{\frac{\mathbf{L}}{2}} \mathcal{L}\, dx\, dy\, du\, dv\ ,$$

$$\mathcal{L} = |\sqrt{(x-ul)^2+(y-vl)^2} - l\sqrt{u^2+v^2+4}|\ ,$$

and

$$E(\mathcal{R}) = \frac{2l}{3\delta^2}\left(\sqrt{(\delta^2+4)^3} - 8\right)\ .$$

4.7 Discussion of Man-in-the-middle Attack

For Schemes 2∼5, the adversary may launch a special man-in-the-middle attack using a long-range out-of-band channel. Consider the following example. Suppose that Alice intends to spend a token $\langle m, \sigma_m \rangle$ at node A. An adversary may eavesdrop on the transmission between Alice and A and overhears $\langle m, \sigma_m \rangle$. He can then relay $\langle m, \sigma_m \rangle$ to another node B using a long-range out-of-band channel. Suppose that A and B chose at least one common witness, say C, and that C receives the token from B prior to receiving it from A. Node C will consider the token from B as a fresh one while that from A as a reuse one. In this way, the adversary can steal Alice's token to access the data at node B.

This attack can be easily defeated by incorporating a *message-specific puzzle* [30] during the token spending process. In particular, to spend token $\langle m, \sigma_m \rangle$ at node A, Alice is required to find a solution X such that the first χ bits of $h(X||m||\sigma_m||A)$ are all zeros, where χ is a system parameter determining the strength of the puzzle. During token spending, Alice sends $X, \langle m, \sigma_m \rangle$ to node A, which will verify both X and σ_m before initiating TRD process. Assuming that the adversary overhears the $\langle m, \sigma_m \rangle$ and intends to spend it at another node B, he needs to find a corresponding solution Y such that the first χ bits of $h(Y||m||\sigma_m||B)$ are all zeros before Alice successfully spends it. As long as the time needed to solve the puzzle is longer than the longest message round-trip time, node C can receive the token from A before receiving it B.

5 A Bloom-Filter Approach to Recording Tokens in Finite Buffers

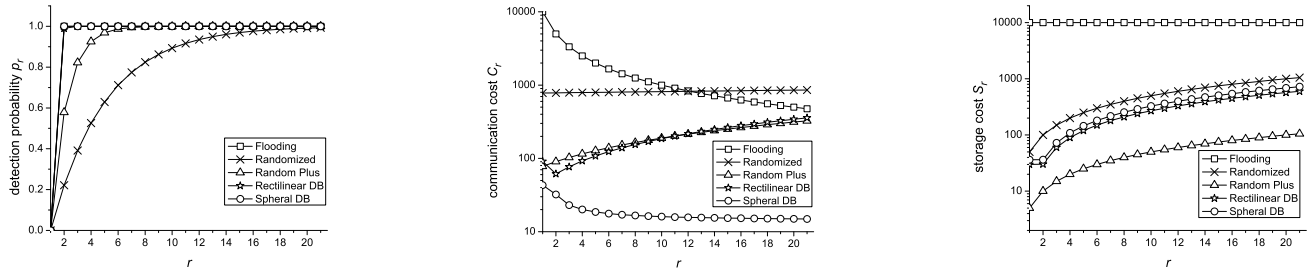
In this section, we extend our DTRD schemes to cope with unlimited number of tokens. In the previous DTRD schemes, each node directly records every token it receives in its buffer. One potential problem might be some sensor nodes' buffers may overflow if too many tokens are stored. To prevent this from happening, the network owner must equip each sensor node with sufficient memory to accommodate the maximum possible number of tokens, which is certainly not cost-effective. Now we introduce a solution for this issue by replacing sensor node buffer with a bloom filter.

A Bloom filter [31] is a space-efficient data structure for representing a set $\mathcal{T} = \{t_i\}_{i=1}^{\omega}$ by a ψ bit vector to support membership checking. The basic idea is to select ϱ independent hash functions $\{h_j\}_{j=1}^{\varrho}$, each with range $\{0, \cdots, \psi-1\}$. For each element $t_i \in \mathcal{T}$, the bits at positions $\{h_j(t_i)\}_{j=1}^{\varrho}$, which are initialized to all zeros, are set to one. For an element b in question, the bits at positions $\{h_j(b)\}_{j=1}^{\varrho}$ are checked. If any of these ϱ bits are zero, b is certainly not belong to $\mathcal{T}$. Otherwise, it is highly likely that b belongs to $\mathcal{T}$. The Bloom filter may also yield false positive, that is, an element is in fact not in $\mathcal{T}$ but all its corresponding bits have been set. Assuming that we have inserted ω elements into the buffer and the output of each of ϱ hash functions is uniformly distributed within $\{0, \cdots, \psi-1\}$, the probability that a particular bit is not set is exactly $(1-1/\psi)^{\varrho\omega}$. It follows that the false positive probability incurred by Bloom filter is

$$\bar{P}_{BF}(\omega) = (1-(1-1/\psi)^{\varrho\omega})^{\varrho} \approx (1-e^{-\varrho\omega/\psi})^{\varrho},$$

which attains the minimum value $1/2^{\varrho} \approx (0.6185)^{\psi/\omega}$ when $\varrho = \ln 2 \times \psi/\omega$ [31]. Since ϱ should be integer in practice, we should choose $\varrho = \lfloor \ln 2 \times \psi/\omega \rfloor$ or $\varrho = \lceil \ln 2 \times \psi/\omega \rceil$, depending on which one results in a smaller false positive probability.

Adopting the Bloom filter can significantly reduce the storage cost of our DTRD schemes or enable each sensor node to store an unlimited number of tokens at the price of slightly increasing false positives. However, the storage cost for each scheme is then a constant which do not vary with the number of tokens stored. In what



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

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

follows, we evaluate the impact of adopting the Bloom filter from two different angles. First, given the same detection probability and false positive probability, how large should the ψ be for each DTRD scheme. Second, given the same detection probability and ψ, how does the false positive probability increase with the number of tokens stored.

To fairly compare the DTRD schemes, we assume that the network owner has issued M tokens, and that each token use generates β witnesses distributed in the witness region uniformly at random. We also assume that all five DTRD schemes use a Bloom filter of the same size ψ, and have the sufficiently high detection probability for the first reuse attempt, e.g., $p_2 > 0.99$.

After M tokens have been spent in the network, the number of tokens stored at any node in the witness region is a random variable following binomial distribution $\mathcal{B}(M, \beta/N_w)$, where N_w is the number of nodes in the witness region. Here $N_w = N$ for Schemes 1∼4, and $N_w = \pi(\mathbf{R_o}^2 - \mathbf{R_i}^2)N/\mathbf{L}^2$ for Scheme 5.

Since M is large in practice, the above binomial distribution can be approximated by normal distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = M\beta/N_w, \sigma^2 = M\beta(N_w - \beta)/N_w^2$. In addition, since the false positive probability monotonically increases with the number of tokens stored in the Bloom filter for a fixed buffer size ψ and the number ϱ of hash functions, the network owner can choose optimal ϱ for $\omega = \mu + \alpha\sigma$, where α is a system parameter. For example, if $\alpha = 3$, then 99.85% nodes will store less than $\mu + 3\sigma$ tokens, which means less than 0.15% nodes will have a false positive probability higher than $(1 - e^{-\varrho(\mu+3\sigma)/\psi})^{\varrho}$.

6 Performance Evaluation

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

6.1 Numeric Results

We assume a square sensor field of 3000×3000 units within which 10000 nodes are uniform randomly distributed. The transmission range of each node is 100 units, which ensures an almost fully connected network [32]. The average number of hops between any two nodes is $\bar{L} \approx 15$, which can be easily calculated according to [33]. Table 1 lists other default parameters used in the evaluation. Under the default configuration, Monte-Carlo simulations indicate $E(\mathcal{L}) = 702$ units and $E(\mathcal{R}) = 578$ units, where $\mathcal{L}$ and $\mathcal{R}$ are the distance between a random point and a random projected circle and the radius of a random projected circle, respectively.

TABLE 1
Default Evaluation Parameters

Para.	Val.	Para.	Val.	Para.	Val.	Para.	Val.
N	10000	D_L	3000	D_W	3000	R	100
L	15	$\mathbf{L}$	3000	ν	20	l	150
M	10^6	θ	100	α	3	ψ	10^6

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 6.2.

Fig. 4(a) compares the TRD probabilities of the five 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 two double ruling schemes Scheme 4 and Scheme 5 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 for a few times (e.g., 8 times).

Fig. 4(b) compare their communication cost in $\log 10$. We can see that Scheme 1 has much higher communication and cost than Scheme 3, Scheme 4 and Scheme 5 for similar TRD probabilities, which make it less suitable for large-scale sensor networks. In addition, Scheme 5 has the lowest communication cost, due to the fact that its communication cost is bounded by $\mathcal{C}_1$. In addition, Scheme 4 outperforms Scheme 3 in the communication cost for small r values (e.g., $r \leq 8$).

Fig. 4(c) compare their storage costs in $\log 10$ scale. We can see that Scheme 1 has much higher storage costs than all the other schemes. In addition, Scheme 3 has the lowest storage cost, followed by Scheme 4 and Scheme 5.

6.2 Simulation Results

We also did simulations using the network configurations in Section 6.1, 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. 5∼7 is the average of the results of spending a random token at 100 random nodes.

6.2.1 The impact of r

Fig. 5 shows the simulation results corresponding to the numerical results in Fig. 4. We assumed in the theoretical analysis that witnesses are mutually different for Scheme 2 and Scheme 3, 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 of Scheme 2 and Scheme 3 are slightly higher than the corresponding simulated ones. In addition, the communication costs of Scheme 4 and Scheme 5 are slightly higher than their numerical counterparts for the following two reasons. First, in Scheme 4 there may be multiple nodes on a replication curve overhearing and responding to a TRD request (cf. Section 4.5) in the simulations in contrast to the single node assumed in the theoretical analysis. Second, the actual number of hops between two nodes may be larger than the ratio between the distance and the transmission range assumed in the analysis. It is thus 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.

6.2.2 The impact of β

Fig. 6 shows 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 Schemes 1, 4 and 5 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 both the chosen witnesses and 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 those of Scheme 4 and Scheme 5 with $\beta = 20$, in which case it has larger communication and storage costs.

6.2.3 The impact of θ

Fig. 7 shows the impact of the number θ of compromised nodes on the TRD capabilities of the five 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 four 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 and Scheme 5 can still detect the first TR attempt with probability 0.97 and 0.99, respectively, which are higher than the numerical results 0.8 and 0.94. The reason is that there might be more intersection nodes than that assumed in the theoretical analysis. These results show that our DTRD schemes have strong resilience against node compromise.

6.2.4 The impact of the Bloom filter

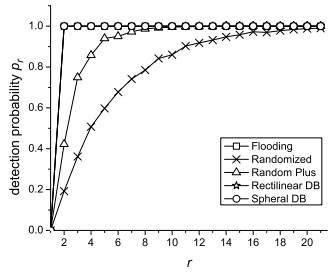
Fig. 8 and Fig. 9 show the impact of the Bloom filter. We assume that none of the sensor nodes are compromised and that p_2s of all five DTRD schemes are higher than 0.99, which is true for Scheme 1, 4 and 5. For Schemes 2 and 3, β is set 215 and 12, respectively, such that $p_2 > 0.99$. Then we compute the corresponding μ, σ, ω and ϕ according to Section 5. In the simulation, we choose the ϱ independent hash functions as $h_i(m) = h(i||m) \mod \psi$, where h is SHA-1. We first insert $\mu + 3\sigma$ random tokens into the Bloom filter, and then try 10000 different tokens to obtain the false positive probabilities.

Fig. 8 shows that given the same buffer size and false positive, Scheme 1 can support the fewest tokens, followed by Scheme 2, Scheme 5, Scheme 4 and Scheme 3. Similarly, Fig. 9 shows that given the same buffer size and number of tokens, Scheme 1 requires the largest buffer followed by Scheme 2, Scheme 5, Scheme 4 and Scheme 3.

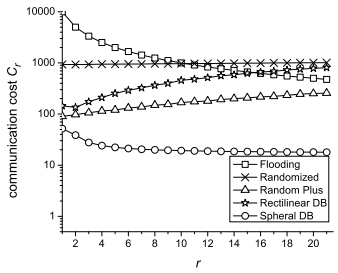
6.3 Discussion

We summarize the performance evaluation as follows.

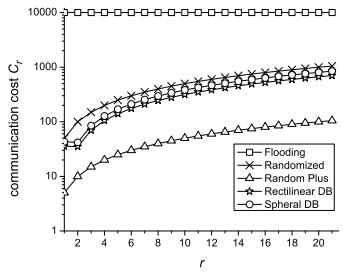
- Scheme 1 can detect any TR attempt with probability 1 with high communication and storage costs.
- Scheme 2 has very poor TRD capability or incurs significant communication and storage costs to achieve a satisfactory TRD probability.
- Scheme 3 has very good TRD capability with reasonable communication 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.
- Scheme 5 has very good TRD capability with very low communication and reasonable storage costs increasing with the field size and the number of successful token reuses.



(a) detection probability p_r

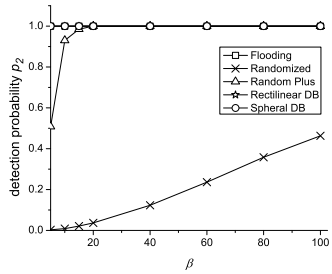


(b) communication cost $\mathcal{C}_r$

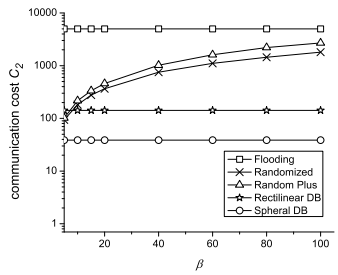


(c) storage cost $\mathcal{S}_r$

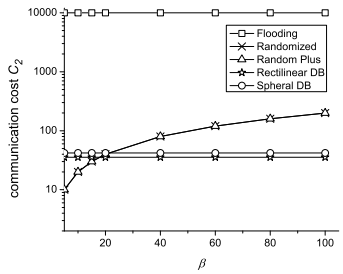
Fig. 5. Simulation result: p_r/$\mathcal{C}_r$/$\mathcal{S}_r$ vs. r.



(a) detection probability p_2



(b) communication cost $\mathcal{C}_2$



(c) storage cost $\mathcal{S}_2$

Fig. 6. Simulation result: p_r/$\mathcal{C}_r$/$\mathcal{S}_r$ vs. β.

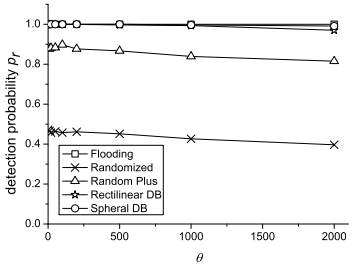


Fig. 7. Detection probability p_r vs. θ.

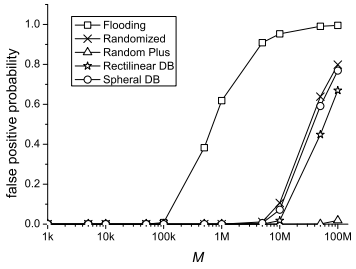


Fig. 8. False positive probability vs. M.

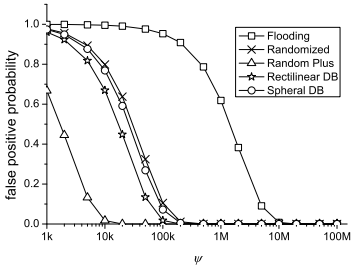


Fig. 9. False positive probability vs. ψ.

In practice, Scheme 5 may be the best choice for large-scale sensor networks, followed by Scheme 4 and Scheme 3. The numerical and simulation results confirm that DP^{2}AC is a very practical and trustworthy solution for sensor networks.

7 Additional Related Work

Besides the related work introduced in Section 1, our work is related to the schemes [34], [35], [36] for detecting node replicates. Among them, [34] assumes the closest network model to ours, so we briefly compare our work with [34] here. In [34], each node generates a location claim which will be forwarded to a set of witnesses selected by its neighbors. Whenever a witness receives a conflicting claim, it considers the corresponding node subject to replication attacks and will revoke it via network-wide broadcasting. Node-replicate detection is thus similar to token-reuse detection in our work. Our DTRD schemes differ significantly from [34] in many aspects. First, our target application is distributed privacy-privacy access control for which DTRD is an essential component, while [34] is for detecting node replicates. Second, in our DTRD schemes, each node at which a token is spent selects the witness set of a constant size. In contrast, in [34], each neighbor of a node selects the witnesses with some probability and the total number of witnesses for a given node is thus a random variable. Third, if there is no compromised node, our best DTRD schemes, i.e., Schemes 4 and 5, can achieve detection probability 1, while the advanced randomized multicast and line-selected multicast schemes in [34] can only detect node replication probabilistically. However, it is fair to say that our DTRD schemes can be applied to

detect node replicates after minor modifications, and the key techniques [34] can also be adapted to detect token reuses. The further exploitation along this direction is beyond the scope of this paper.

Privacy-preserving data aggregation in sensor networks (e.g., [37], [38]) is another active line of research aiming at providing users with some statistics aggregates (e.g., SUM and MAX) of the sensed data without disclosing each individual sensor datum. It is orthogonal to our work in this paper as it assumes a different data access model.

8 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.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants CNS-0716302/1122697, CNS-0844972 (CAREER), CNS-0831963, and CNS-1117811. The authors would like to thank Dr. Jie Gao and Dr. Rik Sarkar for sharing the simulation code of the Double Rulings scheme. The authors would also like to thank anonymous reviewers for their constructive comments.

References

[1] R. Zhang, Y. Zhang, and K. Ren, "DP^{2}AC: Distributed privacy-preserving access control in sensor networks," in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[2] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: A two tier sensor storage architecture using interval skip graphs," in *ACM SenSys'05*, San Diego, California, USA, Nov. 2005, pp. 39–50.

[3] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *IEEE SECON '07*, San Diego, CA, USA, June 2007, pp. 203–212.

[4] B. Sheng, Q. Li, and W. Mao, "Data storage placement in sensor networks," in *ACM MobiHoc'06*, Florence, Italy, May 2006, pp. 344–355.

[5] 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.

[6] H. Wang and Q. Li, "Distributed user access control in sensor networks," in *DCOSS'06*, San Francisco, CA, June 2006, pp. 305–320.

[7] D. Liu, "Efficient and distributed access control in sensor networks," in *DCOSS '07*, Santa Fe, New Mexico, USA, June 2007.

[8] 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.

[9] 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.

[10] 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.

[11] ORION, http://www.joiscience.org/ocean_observing/advisors.

[12] NOPP, http://www.nopp.org/.

[13] IOOS, http://www.ocean.us/.

[14] K. Ren, W. Lou, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1373–1384, July 2006.

[15] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology - Crypto '82*. Springer-Verlag (1983), 1982, pp. 199–203.

[16] I. Osipkov, E. Y. Vasserman, N. Hopper, and Y. Kim, "Combating double-spending using cooperative P2P system," in *ICDCS'07*, Toronto, Canada, June 2007.

[17] J.-H. Hoepman, "Distributed double spending prevention," in *15th Int. Workshop on Security Protocols*, Brno, Czech Republic, Apr. 2007.

[18] L. Hu and D. Evans, "Localization for mobile sensor networks," in *ACM MOBICOM'04*, Philadephia, PA, Sep./Oct. 2004, pp. 45–57.

[19] Y. Chen, K. Kleisouris, X. Li, W. Trappe, and R. Martin, "A security and robustness performance analysis of localization algorithms to signal strength attacks," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 2:1–2:37, Feb. 2009.

[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] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," *Notices of the American Mathematical Society (AMS)*, vol. 46, pp. 203–213, 1999.

[22] 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.

[23] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors (short paper)," in *ICICS'06*, 2006, vol. 4307, pp. 519–528.

[24] C. Livadas and N. Lynch, "A reliable broadcast scheme for sensor networks," Technical Report MIT-LCS-TR-915, MIT CSAIL, 2003.

[25] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *DIALM'99*, Seattle, Washington, Aug. 1999, pp. 48–55.

[26] ——, "Routing with guaranteed delivery in ad hoc wireless networks," *Wirel. Netw.*, vol. 7, no. 6, pp. 609–616, Nov. 2001.

[27] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," in *ACM MOBICOM'06*, Los Angeles, California, USA, Sep. 2006, pp. 286–297.

[28] I. Stojmenović, D. Liu, and X. Jia, "A scalable quorum-based location service in ad hoc and sensor networks," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 1, no. 1, pp. 71–94, Feb. 2008.

[29] B. Nath and D. Niculescu, "Routing on a curve," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 155–160, 2003.

[30] P. Ning, A. Liu, and W. Du, "Mitigating dos attacks against broadcast authentication in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 1, pp. 1–31, Jan. 2008.

[31] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Comm. ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[32] 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.

[33] 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.

[34] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *IEEE S&P'05*, Oakland, CA, May 2005, pp. 49–63.

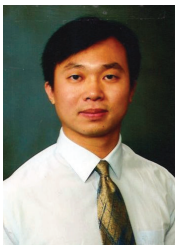
[35] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in *ACSAC'07*, Dec. 2007, pp. 257–267.

[36] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *ACM MobiHoc'07*, Sep. 2007, pp. 80–89.

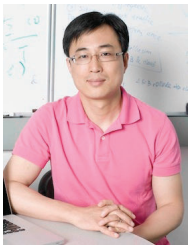
[37] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *IEEE INFOCOM'07*, Anchorage, Alaska, USA, May 2007, pp. 2045–2053.

[38] W. He, H. Nguyen, X. Liu, K. Nahrstedt, and T. Abdelzaher, "iPDA: An integrity-protecting private data aggregation scheme for wireless sensor networks," in *MilCom'08*, San Diego, CA, Nov. 2008, pp. 1–7.

Rui Zhang [S'09] received the B.E. in Communication Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2001 and 2005, respectively. He is currently a Ph.D. student in School of Electrical, Computer, and Energy Engineering at Arizona State University. His research interests are network and distributed system security, wireless networking, and mobile computing. He was a software engineer in UTStarcom Shenzhen R&D center from 2005 to 2007.



Yanchao Zhang [S'03-M'06] received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications, China, in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications, China, in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is currently as an Associate Professor in School of Electrical, Computer, and Energy Engineering at Arizona State University. Before ASU, he was an Assistant Professor of Electrical and Computer Engineering at New Jersey Institute of Technology from 2006 to 2010. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is (was) an Associate Editor of IEEE Transactions on Vehicular Technology, a Feature Editor of IEEE Wireless Communications, a Guest Editor of IEEE Wireless Communications Special Issue on Security and Privacy in Emerging Wireless Networks in 2010, and a TPC Co-Chair of Communication and Information System Security Symposium, IEEE GLOBECOM 2010. He received the NSF CAREER Award in 2009.



Kui Ren [SM'11] is currently an Assistant Professor of Electrical and Computer Engineering Department at the Illinois Institute of Technology. He received his Bachelor's and Master's Degrees from Zhejiang University and a PhD degree from Worcester Polytechnic Institute. Kui's research expertise includes Cloud Computing & Security, Wireless Security, and Smart Grid Security. His research is supported by NSF (TC, NeTS, CSR, NeTS-Neco), DoE, AFRL, and Amazon. He is a recipient of National Science Foundation Faculty Early Career Development (CAREER) Award in 2011. Kui serves as an associate editor for IEEE Wireless Communications and IEEE Transactions on Smart Grid. Kui is a senior member of IEEE and a member of ACM.