Understanding Node Capture Attacks in User Authentication Schemes for Wireless Sensor Networks

Chenyu Wang, Ding Wang®, Yi Tu®, Guoai Xu®, and Huaxiong Wang®

Abstract—Despite decades of intensive research, it is still challenging to design a practical multi-factor user authentication scheme for wireless sensor networks (WSNs). This is because protocol designers are confronted with a long-standing “security versus efficiency” dilemma: sensor nodes are lightweight devices with limited storage and computation capabilities, while the security requirements are demanding as WSNs are generally deployed for sensitive applications. Hundreds of proposals have been proposed, yet most of them have not been found to be problematic, and the same mistakes are repeated again and again. Two of the most common security failures are regarding smart card loss attacks and node capture attacks. The former has been extensively investigated in the literature, while little attention has been given to understanding the node capture attacks. To alleviate this undesirable situation, this article takes a substantial step towards systematically exploring node capture attacks against multi-factor user authentication schemes for WSNs. We first investigate the various causes and consequences of node capture attacks, and classify them into ten different types in terms of the attack targets, adversary’s capabilities and vulnerabilities exploited. Then, we elaborate on each type of attack through examining 11 typical vulnerable protocols, and suggest corresponding countermeasures. Finally, we conduct a large-scale comparative measurement of 61 representative user authentication schemes for WSNs under our extended evaluation criteria. We believe that such a systematic understanding of node capture attacks would help design secure user authentication schemes for WSNs.

Index Terms—User authentication, node capture attacks, wireless sensor networks

1 INTRODUCTION

The past ten years have witnessed the prosperity and development of wireless sensor networks. As the elementary infrastructure of Internet of Things, WSNs are widely used in smart homes [1], public safety [2], personal health [3] and intelligent transportation systems [4]. A WSN is an ad-hoc network consisting of a large number of sensor nodes which are connected by wireless communication. These sensor nodes can collaboratively monitor information from network coverage area [5], and typically external parties are allowed to access the real-time data in sensor nodes to acquire the status of the monitoring entity [6], [7]. As such, it is critical that the sensitive data are not accessed by malicious adversaries. Therefore, a well-designed user authentication method is necessary.

Generally, there are three factors used for authenticating a person: something only she knows, such as a password [8]; something she has, such as a smart card; something she is, such as a biometric trait [9]. Due to their simplicity and convenience, password-based authentication protocols get quite popular [10]. Smart card and biometric factors are usually added to password-based protocols as a way for increasing security [1], [11]. A protocol which combines at least two factors is called a multi-factor user authentication protocol. It is typically used for security-crucial systems, such as wireless sensor networks as shown in Fig. 1. In this authentication models, three participants are included: 1) A set of users U, who may want to access the real-time data from a sensor node, and 2) A large number of distributed sensor nodes SN, which are deployed to detect, monitor and collect data, and may help to process the data; 3) The gateway GWN, who acts as a controller, provides a registration service, and is a communication bridge between users and sensor nodes. Though our results can be also applied to multi-gateway environments, this paper primarily focuses on the single-gateway architecture as shown in Fig. 1. Besides, unless otherwise specified, the figures, tables and various conclusions in this article are for multi-factor user authentication in WSNs.

The request for a user authentication protocol that ensures the security of communication and avoids eavesdropping by adversaries, has resulted in a large number of proposals. However, designing a multi-factor user authentication scheme for WSNs is full of challenges due to the fact that the protocol designer is confronted with a powerful adversary, resource-
To reveal the difficulties in designing a multi-factor user authentication scheme for WSNs, we revisit dozens of typical schemes and identify their security flaws. For a concrete grasp, we show the result in Fig. 2, illustrating the development history of multi-factor user authentication protocols for WSNs. It can be seen that most proposals have been found insecure or unable to provide certain important security attributes. Particularly, most schemes are unable to resist offline dictionary attacks or node capture attacks. Among these examined schemes, only two are secure against node capture attacks.

To repair offline dictionary attacks, much effort has been made. Notably, Ma et al. [12] showed that the public-key algorithm is indispensable; Wang et al. [13] introduced a technique integrating “fuzzy-verifier” and “honey-words” to settle the conflict between resisting against offline dictionary attacks and detecting typos timely. In contrast, little attention has been paid to node capture attacks. In most cases, node capture attacks are mentioned as a security threat, but their actual attacking process and consequences are overlooked, and there is a lack of systematic investigation of node capture attacks.

### 1.1 Node Capture Attacks

Around 2000, Carman et al. [14] and a number of researchers [15], [16], [17] pointed out that the adversary can physically acquire the data of some sensor nodes, because they are usually left in unattended or hostile environments and it costs too much to equip them with tamper-resistant hardwares in view of their large-scale development. In 2005, Benenson et al. [18] for the first time introduced node capture attacks into remote user authentication schemes. They pointed out that the adversary can compromise some of sensor nodes and carry out a series of subsequent attacks. After that, node capture attacks begin to be accepted as a practical attack against user authentication schemes for WSNs, and many new proposals are designed to resist against this attack.

One notable attempt, initiated by Vaidya et al. [19] in 2010, is to identify the weaknesses in previous schemes [20], [21], [22] against node capture attacks and design a new secure version. Unfortunately, this scheme was later shown that it does not provide forward secrecy, and is insecure against smart card loss attacks and node capture attacks. In 2012, Vaidya et al. [23] proposed a hash-based scheme which is claimed to be secure against node capture attacks.

However, in 2014, Kim et al. [24] pointed out that Vaidya et al.’s scheme [23] is unable to resist node capture attacks: if the adversary gets the private key of a sensor node, she can forge the message that is sent by the gateway to users. They then proposed a scheme which is claimed to be secure against node capture attacks and other known attacks. Nevertheless, their security claims were invalidated by Chang et al. [25], who showed that a legitimate user can get sensor node’s private key and carry out subsequent attacks. Therefore, Chang et al. proposed a new enhanced version, and proved their scheme is secure against node capture attacks. However, their scheme was demonstrated by Park et al. [26] that it does not provide forward secrecy if the adversary obtains the private key of a sensor node, and thus it is not secure against node capture attacks. Recently, in 2017, Srinivas et al. [27] presented a temporary-certificate-based user authentication...
scheme for WSNs with lightweight operations. Particularly, they proved that their scheme is secure against node capture attacks. Later, Wang et al. [6] revealed that in Srinivas et al.’s scheme [27], once a sensor node is compromised, the adversary is able to compute previous session keys that are associated to this sensor node. Thus, the scheme in [27] is unable to resist node capture attacks again.

As said above, node capture attacks have been considered as a practical attack against user authentication schemes for WSNs. More and more schemes take the resistance to node capture attacks as an attribute that should be satisfied [6], [23], [27], but most schemes still suffer from this threat. Moreover, when assessing the security of multi-factor user authentication schemes for WSNs, node capture attacks are usually included in the criterion “resistance to known attacks” (see [6], [28]). In a nutshell, The harmfulness of node capture attacks have not been well recognized and a systematic investigation is still lacking.

1.2 Motivations and Contributions

Generally, sensor nodes are deployed in unattended or hostile environments, and its large-scale deployment makes it too costly to equip them with tamper-resistant hardwares. Hence, sensor nodes are susceptible to be captured by adversaries, resulting in typical node capture attacks in user authentication schemes for WSNs.

1) Although some recent work takes into account of node capture attacks, as mentioned above, most of them still cannot resist against node capture attacks, and they are caught in a “break-fix-break-fix” circle. The main reason for this undesirable situation is a lack of systematic investigation on node capture attacks.

2) Moreover, the damaging effects of node capture attacks are underestimated. According to our observation and examples in later sections, besides triggering the leakage of previous session keys [6], node capture attacks may enable adversaries to trace user activities, impersonate users, manipulate not only the compromised sensor nodes but also other nodes, and even break the security of the entire system.

In all, node capture attacks have become one of the most urgent and prevalent issues to be addressed in the design of a secure user authentication scheme for WSNs, and they would have a huge impact on the security of user authentication schemes. Understanding node capture attacks and summarizing their causes and consequences can help to design a secure authentication scheme that can resist against this kind of attack, which motivates us to conduct a systematic investigation on node capture attacks.

As far as we know, this is the first in-depth exploration on node capture attacks in the field of user authentication schemes for WSNs. Towards our goal, we first define the adversary model based on Wang et al.’s work [6]. Then, we put forward a detailed and thorough evaluation criteria for multi-factor user authentication schemes for WSNs. We achieve this by combining the merits of the widely accepted evaluation criteria [6], [13] and including the effects of node capture attacks. Note that, unlike [6], [13], where they include the resistance to node capture attacks in the criterion C5 “resistance to known attacks”, we propose a separate criterion “resistance to node capture attacks”. This additional criterion is indispensable to understand and evaluate the security of multi-factor user authentication schemes for WSNs due to the prevalent feature and damaging effects of node capture attacks.

Then, with intensive experience on analyzing about ninety user authentication schemes for WSNs, we figure out the various causes and consequences of node capture attacks, and classify them into ten types in terms of the attack targets, adversary’s capabilities and vulnerabilities exploited. We explain each type of attack through examining typical vulnerable schemes, and propose corresponding countermeasures. For example, in Fan et al.’s scheme [29], due to the inappropriate distribution of sensor nodes’ private keys, all sensor nodes share a same private key with the gateway. We show that an adversary who compromises the sensor node SNj can obtain the private keys of all sensor nodes, resulting to sensor node impersonation threat. To deal with this attack, we recommend to use h(IDSNj|x) as SNj’s private key, where x is a long-term secret key, IDSNj is SNj’s identity.

Finally, according to our taxonomy of node capture attacks, we naturally improve our evaluation criteria for multi-factor user authentication schemes for WSNs by expanding the criterion “resistance to node capture attacks” into ten types. We then perform a large-scale assessment of 61 multi-factor user authentication schemes for WSNs under our expanded criteria set. Among those schemes, only two are secure against node capture attacks, indicating the difficulty in designing node-capture-attack resistant user authentication schemes for WSNs along the way. Fortunately, our work provides a better understanding of node capture attacks, and we believe that this work would facilitate the design of secure user authentication schemes for WSNs that is resistant to node capture attacks. In brief, our contributions are summarized as follows:

1) We investigate the root causes and consequences of node capture attacks against user authentication schemes for WSNs, and classify them into ten different types in terms of the attack targets, adversary capabilities and vulnerabilities exploited. As far as we know, we are the first to provide a taxonomy of node capture attacks.

2) We elaborate on each type of node capture attacks through examining a corresponding typical vulnerable scheme, and propose corresponding countermeasures.

3) Finally, based on our taxonomy of node capture attacks, we extend our evaluation criteria, and perform a large-scale assessment of 61 user authentication schemes for WSNs under the expanded criteria.

1.3 Paper Organization

The remaining sections are organized as follows. In Section 2, we describe the adversary model, evaluation criteria and notions used in the paper. Section 3 presents a taxonomy of node capture attacks. Section 4 explains each type of node capture attacks using several typical schemes. The countermeasures are given in Section 5. Section 6 gives a large-scale measurement of 61 representative authentication schemes under our extended evaluation criteria. The summary of this paper is given in Section 7.
2 Adversary Model, Evaluation Criteria, and Model of Authentication Process

In this section, we first introduce some notations and a standard model of authentication process for public-key based multi-factor user authentication schemes for WSNs, then define the adversary model and evaluation criteria based on widely accepted frameworks.

2.1 A Generic Model of Authentication Process

Our notations and abbreviations are illustrated in Table 1, and the standard model of authentication process for WSNs is shown in Fig. 3. Note that, this model is recommended by Wang et al. [6], because other models for single-gateway WSNs have some inherent weaknesses.

Table 1: Notations and Abbreviations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ui</td>
<td>user</td>
<td>SNj</td>
<td>sensor node</td>
</tr>
<tr>
<td>Ui</td>
<td>the i\textsuperscript{th} user</td>
<td>SNj</td>
<td>the j\textsuperscript{th} sensor node</td>
</tr>
<tr>
<td>GWN</td>
<td>gateway node</td>
<td>SCI</td>
<td>U\textsubscript{i}'s smart card/device</td>
</tr>
<tr>
<td>A</td>
<td>the adversary</td>
<td>x</td>
<td>GWN's long-term secret key</td>
</tr>
<tr>
<td>GI</td>
<td>identity of Ui</td>
<td>PW\textsubscript{i}</td>
<td>password of Ui</td>
</tr>
<tr>
<td>ID\textsubscript{SN}</td>
<td>identity of SNj</td>
<td>SK</td>
<td>session key*</td>
</tr>
<tr>
<td>X\textsubscript{GU}</td>
<td>secret key of Ui</td>
<td>X\textsubscript{SN}</td>
<td>secret key of SNj</td>
</tr>
<tr>
<td>⊕</td>
<td>bitwise XOR operation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Throughout the paper, the session key is between the user and sensor node.

![Fig. 3. Authentication processes.](image)

1) We name parameters Auth\textsubscript{1}, Auth\textsubscript{2}, Auth\textsubscript{3} and Auth\textsubscript{4} used for verifying the validity of participants as verification parameter, denoted as VP.

2) The shared secret key X\textsubscript{GU} is named as fixed unique secret parameter between GWN and U\textsubscript{i}, denoted as FUSP\textsubscript{GU}/U. Similarly, X\textsubscript{SN} is named as fixed unique secret parameter between GWN and SN\textsubscript{j}, denoted as FUSP\textsubscript{GS}.

3) The shared secret parameter R\textsubscript{GU} is named as temporary unique secret parameter between GWN and U\textsubscript{i}, denoted as TUSP\textsubscript{GU}/U. R\textsubscript{GS} is named as temporary unique secret parameter between GWN and SN\textsubscript{j}, denoted as TUSP\textsubscript{GS}.

4) Parameter r\textsubscript{i} that is chosen by U\textsubscript{i} and critical in computing session keys is named as SK-U-critical parameter, denoted as CP\textsubscript{SK}/U. Parameter r\textsubscript{j} that is chosen by SN\textsubscript{j} and essential in the computation of session keys is named as SK-S-critical parameter, denoted as CP\textsubscript{SK}.

Within a user authentication protocol, if an adversary is able to obtain any of: PW\textsubscript{i}, for some i; X\textsubscript{SN} for some j; or the long-term secret key x, we say that the adversary can fully impersonate their victim (U\textsubscript{i}, SN\textsubscript{j}, or GWN, respectively). Such an attack is called a complete impersonation attack. On the other hand, if the adversary has no such secret information and can only try to manipulate some parties' messages to cheat other parties, we name such an attack as an incomplete impersonation attack.

2.2 Adversary Model and Evaluation Criteria

As the security of a cryptographic scheme cannot be properly evaluated if the adversary model or evaluation criteria is not well defined, we now describe the adversary model and evaluation criteria, tailored to multi-factor user authentication protocols for WSNs in the single-gateway setting.

Our adversary model is adapted from the one in [6] and is defined in Table 2. Note that Wang et al.'s criteria set [6] only considers the two-factor authentication scenario. Therefore, we adjust C3 of [6] so that it captures the three-factor scenario considered in this work. Also, we remove the adversary’s ability in multi-gateway setting in C7 as we only consider the single-gateway environment.

Our evaluation criteria is adapted from the state-of-the-art evaluation frameworks [6], [13] and the traditional one [51], and it is illustrated in Table 3. More specifically, following [51], we divide the criteria into two levels: the ideal attributes and security requirements. The former deals with various attributes that an ideal user authentication scheme should provide, and focuses on the usability of the

1. When assessing forward secrecy, this key can be extracted from the network in the registration phase, it interacts with the gateway node and evaluates the information that helps to compute the session key. It is assumed that this key is well protected and cannot be extracted from GWN's database.

...
The latter specifies requirements that a scheme should satisfy to be served as a secure one. Following [6], [13], we remove redundancies in the criteria of [51] and form our 12 independent criteria. Inspired by [6], [13], we separate node capture attacks from "the known attacks" and propose our criterion S6 (resistance to node capture attacks), taking into account the prevalent features and damaging effects of node capture attacks. More specifically, the reason why we propose an independent criteria for node capture attacks, are consistent with Wang et al. [13], where they separate smart card loss attacks from the criterion C5 "resistance to known attacks" due to the destructive effects of smart card loss attacks. Another difference of our criteria from [6], [13], [51] is that we specify the adversary’s capabilities for each criterion. For the criteria under ideal attributes, we evaluate them from the functional perspective rather than from the attacking view. For the criteria under security requirements, we specify the adversary’s capabilities in Table 3.

**Remark 1.** From the above comparison, we can see that the most important difference between our criteria and existing criteria [6], [13], [51] is that our criteria proposes a separate criterion S6 “resistance to node capture attacks”, yet this criterion is included in the criterion “resistance to known attacks” in existing criteria. Furthermore, as shown in Section 6, our criteria framework will further divide S6 into ten sub-criteria based on our analysis results of node capture attacks. This difference is the main reason why it seems that our criteria becomes more complex than others. However, we think this complexity is necessary and it will make our criteria more concrete and decisive to be employed. In these existing criteria [6], [13], [51], the attack scenarios where the adversary simultaneously compromise the victim’s smart card and several sensor nodes, cannot be captured. Besides, our criteria framework allows the designers to assess the scheme more objectively and easily. For example, in previous criteria framework, if the protocol designer wants to assess whether their scheme can satisfy the criterion “resistance to known attacks”, she needs to assess whether their scheme can resist to node capture attacks. But how to achieve this? Before our work, they need to try various possible attack scenarios, which either may cost more time to assess or ignore some important attack scenarios. Fortunately, the ten sub-criteria (as shown in Section 6) of our criteria framework provide a structured, actionable and concrete reference for protocol designers to systematically evaluate whether their scheme can resist against node capture attacks.

**Remark 2.** All authentication schemes are assessed from two aspects: (1) The security and functionality under a widely-accepted criteria framework; (2) The performance, such as computational cost and storage cost. The former captures the security and functionality requirements, and the latter specifies requirements that a scheme should satisfy to be served as a secure one. Following [6], [13], we remove redundancies in the criteria of [51] and form our 12 independent criteria. Inspired by [6], [13], we separate node capture attacks from "the known attacks” and propose our criterion S6 (resistance to node capture attacks), taking into account the prevalent features and damaging effects of node capture attacks. More specifically, the reason why we propose an independent criteria for node capture attacks, are consistent with Wang et al. [13], where they separate smart card loss attacks from the criterion C5 “resistance to known attacks” due to the destructive effects of smart card loss attacks. Another difference of our criteria from [6], [13], [51] is that we specify the adversary’s capabilities for each criterion. For the criteria under ideal attributes, we evaluate them from the functional perspective rather than from the attacking view. For the criteria under security requirements, we specify the adversary’s capabilities in Table 3.

### Table 2
**Capabilities of the Adversary**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>A can control messages transmitted among U, SN and GWN.</td>
</tr>
<tr>
<td>C2</td>
<td>A can offline enumerate all items in the Caresian product of identity space and password space $P_d \times P_p$ within polynomial time, or get U’s identity only when evaluating the scheme’s security.</td>
</tr>
<tr>
<td>C3</td>
<td>To a n-factor (n = 2 or 3) user authentication scheme, A can compromise following n – 1 factors: (1) password; (2) data in smart card; (3) biometric.</td>
</tr>
<tr>
<td>C4</td>
<td>A can acquire previous session keys between U and SN.</td>
</tr>
<tr>
<td>C5</td>
<td>A can learn GWN’s secret key(s) when assessing the system’s initial failure.</td>
</tr>
<tr>
<td>C6</td>
<td>A can break some SN, i.e. extracting the sensitive data stored in SN, and controlling the broken sensor node to join the communication of U and GWN.</td>
</tr>
<tr>
<td>C7</td>
<td>A may register to be a legitimate user. Only when assessing the security of the users’ passwords in the registration phase, A can also be the administrator of the gateway.</td>
</tr>
</tbody>
</table>

1: The seven capabilities are not all necessary to Table 4 where C4 and C5 are not mentioned, because these two capabilities have no inherent relevance to the taxonomy of node capture attacks. However, we include all seven capabilities here for completeness.

2: Since the multi-gateway environment is not our focus, we omit the part about it. Furthermore, we highlight the security threat of the administrator of the gateway in the registration phase.

### Table 3
**Evaluation Criteria**

<table>
<thead>
<tr>
<th>Short term</th>
<th>Definition in WSNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Password Friendly*</td>
</tr>
<tr>
<td>D2</td>
<td>Key Agreement</td>
</tr>
<tr>
<td>D3</td>
<td>No clock synchronization</td>
</tr>
<tr>
<td>D4</td>
<td>Mutual Authentication</td>
</tr>
<tr>
<td>D5</td>
<td>No Password Verifier table</td>
</tr>
<tr>
<td>S1</td>
<td>User Anonymity</td>
</tr>
<tr>
<td>S2</td>
<td>No Password Exposure</td>
</tr>
<tr>
<td>S3</td>
<td>Forward Secrecy</td>
</tr>
<tr>
<td>S4</td>
<td>Resistance to Known Attacks</td>
</tr>
<tr>
<td>S5</td>
<td>Resistance to Smart Card Loss Attacks</td>
</tr>
<tr>
<td>S6</td>
<td>Resistance to Node Capture Attacks</td>
</tr>
</tbody>
</table>

*: An ideal attribute is assessed from the functional perspective rather than an attack.

**: The criterion “Timely Typo Detection” in [6] is included in D1 here, as a scheme providing local-change-password can timely detect typos too. Note that, we say that A breaks S5 and S6 only when A conducts the attack with the help of compromised smart card and sensor node, respectively.
TABLE 4
A Taxonomy of Node Capture Attacks^1

<table>
<thead>
<tr>
<th>Type</th>
<th>Attack Target</th>
<th>A’s Capabilities</th>
<th>Vulnerabilities Exploited</th>
<th>Attack Consequences</th>
<th>Attack Scale^2</th>
<th>Refer.</th>
<th>Vulnerable Schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Session key</td>
<td>× × × × ×</td>
<td>the issue of forward secrecy</td>
<td>get previous SK of SNj</td>
<td>(a)→(c)</td>
<td>Sec. 4.1</td>
<td>[21], [30]</td>
</tr>
<tr>
<td>II</td>
<td>Users</td>
<td>× × × × ×</td>
<td>insecure transmission of U’s unique secret parameter</td>
<td>get FUSP <em>g/U</em></td>
<td>(a)→(b)→(c)→(d)</td>
<td>Sec. 4.2</td>
<td>[29], [31]</td>
</tr>
<tr>
<td>III</td>
<td>Users</td>
<td>× × × × ×</td>
<td>same to offline dictionary attacks in distributed system</td>
<td>get users’ password</td>
<td>(a)→(b)→(c)→(d)</td>
<td>Sec. 4.4</td>
<td>[30], [34]</td>
</tr>
<tr>
<td>IV</td>
<td>Users</td>
<td>× × × × ×</td>
<td>1. Inappropriate distribution of SN’s secret key, or</td>
<td>break user anonymity</td>
<td>(a)→(b)→(c)→(d)</td>
<td>Sec. 4.5</td>
<td>[37], [38], [39]</td>
</tr>
<tr>
<td>V</td>
<td>Sensor node</td>
<td>× × × × ×</td>
<td>1. Inappropriate distribution of SN’s secret key, or</td>
<td>impersonate SNm</td>
<td>(a)→(b)→(c)→(d)</td>
<td>Sec. 4.7</td>
<td>[40], [42]</td>
</tr>
<tr>
<td>VI</td>
<td>Gateway</td>
<td>× × × × ×</td>
<td>1. Inappropriate distribution of SN’s secret key, or</td>
<td>impersonate SNm</td>
<td>(a)→(b)→(c)→(d)</td>
<td>Sec. 4.8</td>
<td>[45], [46], [47], [48]</td>
</tr>
<tr>
<td>VII</td>
<td>Availability</td>
<td>× × × × ×</td>
<td>1. U’s login request fails to identify the target SN, and</td>
<td>modify session key</td>
<td>(a)→(c), or</td>
<td>Sec. 4.1</td>
<td>[38], [50]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X D × × ×</td>
<td>and all users, it is more destructive.</td>
<td></td>
<td>(a)→(b)→(c)→(d)</td>
<td></td>
<td>[34], [39]</td>
</tr>
</tbody>
</table>

^1 In this table, we assume the adversary has broken the sensor node SNj, Uj is a legitimate user and willing to collude with A, as well as a victim that A tries to attack. SNm (m ≠ j) denotes the sensor nodes that A attempts to attack.

^2 C6', C3' and C1' all are parts of the capabilities C6, C3 and C1, respectively. C6': here we want to emphasize that A extracts the data in Uj’s card (and gets the biometrics). C1': here C1' refers to that A modifies and sends message to the participants (Uj/SNj/GWN). Since C4 and C5 have no influence to the classification, they are not listed.

^3: The capability is required. √: the capability is required. D: whether the capability is required depends on specific attack scenario.

^4: As shown in Fig. 4, it describes the status of the affected entity with the increase in the number of attacks.

^5: It usually happens in the model (b) of Fig. 5.

^6: It usually happens in the model (c) of Fig. 5.

is continuing to be a hot and hard topic and has led to intense research, see [6], [13], [51]. The latter is specific and can well capture the dynamic nature of WSNs. Like existing criteria frameworks, the target of our criteria is to assess the security and functionality of authentication schemes. When WSNs become larger, the factors that affect the security, such as the capabilities to compromise the victim’s smart card, some of sensor nodes and long-term secret key, will not change. Therefore, the scale of WSNs has little impact on the security of authentication schemes. As for the functionality, the criterion “Sound Repairability”, which requires the scheme to support dynamic sensor node addition, is an attribute to support the dynamic nature of WSNs. Therefore, when WSNs become larger, existing criteria frameworks and ours are still workable.

3 A TAXONOMY OF NODE CAPTURE ATTACKS

Based on the analysis of about 90 authentication protocols for WSNs, we investigate the causes and consequences of node capture attacks, and we classify them into ten different types (see Table 4) in terms of the attack targets, adversary’s capabilities and vulnerabilities exploited. As shown in Table 4, the adversary A can achieve different attack consequences and attack scale^2 in terms of different attack targets, adversary’s capabilities and vulnerabilities exploited. The attack targets can be divided into five categories: the session keys, the users, the sensor nodes, the gateway and the availability. The vulnerabilities exploited include insecure parameter transmission, inappropriate parameter distribution, unreasonable design intent, inefficient verification, the issue of offline dictionary attacks and forward secrecy.

Type I and Type II in Table 4 depict the attacks where A breaks the security of session keys with the help of the private key X _SNj_ of compromised sensor node SNj. The difference between them is that: in Type I, A can only calculate previous session keys between the compromised sensor node SNj and all users; in Type II, A can calculate previous session keys between all sensor nodes and all users, it is more destructive. The root cause of Type I is essentially consistent with that of forward secrecy. As for Type II, besides the issue of forward secrecy, the inappropriate distribution of SN’s private key is its another cause.

Type III～Type V in Table 4 represent the scenarios where the adversary with X _SNj_ compromises the security of users. Both Type III and Type IV can be regarded as a user impersonation attack. In Type III, the adversary gets the victim’s fixed unique secret parameter FUSP _g/U_ (X _Uj_ in Fig. 3) to impersonate Uj, due to the insecure transmission (such as “XOR” operation) of user unique secret parameters. In Type IV, the adversary with X _SNj_ and who additionally gets the data stored in a victim’s smart card (and the biometric) and can enumerate the items in the space of password and identity, is able to obtain users’ passwords, and then acts as Uj to engage in the conversations. This attack is an outcome of no (or incorrect) public-key algorithm deployment. Its failure reason is as same as offline dictionary attacks [12].

If the adversary A cannot impersonate the user, then she may try to break the victim’s privacy, and this is the situation in Type V. It contains two cases. In the first case, A can trace users’ activities, but cannot compute their identities. It usually occurs in a temporary-certificate-based authentication protocol where A can trace the victim by manipulating the broken sensor node to seek the link parameter, such as TID, in [37]. In the second case, A can compute users’ identities IDi successfully, because they are transmitted with simple “XOR” operation or designed to be known to sensor nodes.
(e.g., simply transmit ID to sensor nodes in plaintext), and we call this flaw an unreasonable design intent.

In Type VI~Type VIII in Table 4, once the adversary $A$ compromises the sensor node $SN_j$ and gets $X_{SN_j}$, then she also can compromise other sensor nodes $SN_m$ $(m \neq j)$ in different ways. In Type VI, $A$ with $X_{SN_j}$ exploits the insecure transmission or distribution of the private key of sensor nodes $SN_m$ $(m \neq j)$ to obtain the private key and impersonate $SN_m$. In Type VII, due to the insecure transmission of users’ unique secret parameters and GWN’s failure in authenticating $SN$ (usually happen in the communication model of (b) in Fig. 5), $A$ with $X_{SN_j}$ then can impersonate $SN_m$ to users. In Type VIII, since the users’ login requests are first sent to sensor nodes without explicitly designating the target sensor node $SN_m$ (usually happens in the communication model of (c) in Fig. 5), the adversary with $X_{SN_j}$ then can intercept the login request sent to $SN_m$, and act as $SN_j$ to respond the request as the process of the original protocol without being noticed by users. After this attack, the users think a session key is agreed with $SN_m$, but actually with $SN_j$ (i.e., the adversary). Among the three attacks, we say that only Type VI where the adversary becomes $SN_m$ achieves complete impersonation. $A$ in Type VII and VIII tries to disguise as much as possible to deceive $U_i$ and GWN, so only achieves incomplete impersonation.

In Type IX, the adversary who registers as or colludes with a legitimate $U_i$ and exploits the weakness in the insecure transmission of GWN’s long-term secret key, can get the secret key. It makes whole system completely insecure, because GWN’s long-term secret key is employed to compute all secret information of users and sensor nodes.

The last attack Type X is very common in authentication schemes based on our analysis, but it receives little notice. In Type X, the adversary $A$ with $X_{SN_j}$ can modify session keys between users and sensor nodes. Furthermore, it is a progressive attack. If $U_i$ fails to verify part of SK controlled by $SN_j$, i.e., $CP_{SK/U}$, then $A$ with $SN_j$’s private key can tamper the respond message from $SN_j$/GWN to users (i.e., message $M_4$ in Fig. 3), and makes legitimate participants (i.e., all users and $SN_j$) unable to share the same session key, meanwhile the participants authenticate to each other successfully. If besides the problem above, the part of SK controlled by $U_i$ (e.g., $CP_{SK/U}$) is also transmitted insecurely, then $A$ can modify session keys between $U_i$ and all sensor nodes. Type X not only causes usability problems where legitimate parties cannot share a same session key and thus cannot correctly decrypt their interaction messages, but also enable the adversary to compute the same session key with $U_i$.

## 4 Examples of the Ten Types of Node Capture Attacks

To better understand the ten different types of attacks in Table 4, we explain them in detail by using several typical user authentication protocols for WSNs. Note that, to save space, we do not review the original protocols, and we retain the symbols and notations of the original protocols even though they are not the same as those in Table 1.

### 4.1 Node Capture Attack Type I

Type I depicts a practical attack where the adversary with the private key $X_{SN_j}$ of sensor node $SN_j$ can compute the session keys between $SN_j$ and all users. The attack cause of Type I is the same as that of forward secrecy. This section uses Kumari et al.’s scheme [2] to explain this attack.

- **Adversary’s Capability:**
  1. “C1”. Eavesdrop the message between GWN and $SN_j$ in authentication phase to get $\{A_5, A_6, C_2\}$.
  2. “C6”. Get $SN_j$’s private key $SIX_k$.
- **Attack Target**: session key.
- **Attack Consequence**: compute $SN_j$’s previous session keys.
- **Attack Steps**:
  1. Compute $(RU_i^{*}||RG_j^{*}||TS_2^{*}) = A_5 \oplus SIX_k$.
  2. Compute $A_1^{*} = A_6 \oplus h($SID$_i||h(RG_j^{*})||RU_i^{*})$.
  3. Compute $(RS_j^{*}||TS_3^{*}) = C_2 \oplus RG_j^{*}$.
  4. Compute $SK_i = h(A_1^{*}||RU_i^{*}||RS_j^{*})$.

- **Time Complexity**: $O(3T_H)$, where $T_H$ denotes the running time of hash operation. Some lightweight operations like XOR and $||$ are omitted.

- **The Scale of the Attack**: (a)–(c) as shown in Fig. 4. In the beginning, the adversary $A$ compromises the
sensor node $SN_1$ (marked in red circle as shown in (a) of Fig. 4), then exploits messages among $GWN$, $SN_1$ and $U_1$, $A$ can compute previous session keys between $U_1$ and $SN_1$ as above. Similarly, $A$ can compute previous session keys between $U_2$ and $SN_1$. When the number of the attacks is large enough, $A$ can compute previous session keys between $SN_1$ and all users as shown in (c) of Fig. 4.

4.2 Node Capture Attack Type II
As we mentioned above, the issue of forward secrecy and inappropriate distribution of sensor nodes’ private keys result in Type II. In this section, we take Fan et al.’s scheme [29] as an example to describe node capture attack Type II. Note that all sensor nodes and the gateway share the same secret parameter $S_k$ in Fan et al.’s scheme [29].

- Adversary’s Capability:
  (1) “C1”. Eavesdrop the message between $GWN$ and $SN_1$ in authentication phase to get $K$.
  (2) “C6”. Get $SN_1$’s private key $S_k$.
- Attack Target: session key.
- Attack Consequence: compute previous session keys between all sensor nodes and all users.
- Attack Steps:
Step 1. Compute $Key = h(S_k||K)$.
Step 2. Compute $\mathbf{K}_1^0 = S_k^0P$.
Step 3. Compute $\mathbf{K}_2^0 = S^0X$, note that $X$ is a public parameter and can be easily gotten.
Step 4. Compute $\mathbf{M}_1^4 = ID_i \oplus \mathbf{M}_1^4$.
Step 5. Compute $\mathbf{M}_2^1 = \mathbf{M}_1^1 \oplus r_1^4$.
Step 6. Compute $\mathbf{M}_3^1 = h(ID_i||r_1^4) \oplus \mathbf{SID}_m$ (m can be any valid number).
Step 7. Compute $\mathbf{M}_4^1 = h(\mathbf{M}_1^1||r_1^4)\oplus \mathbf{SID}_m$ (m can be any valid number).
Step 8. Compute $\mathbf{M}_5^1 = \mathbf{M}_2^1 \oplus r_0^1$.
Step 9. Compute $\mathbf{M}_6^1 = h(\mathbf{M}_1^1||r_1^4)\oplus \mathbf{SID}_m$ (m can be any valid number).
Step 10. Compute $\mathbf{M}_7^1 = h(\mathbf{M}_1^1||r_1^4)\oplus \mathbf{SID}_m$ (m can be any valid number).
Step 11. Send $\{\mathbf{M}_2^1, \mathbf{M}_4^1, \mathbf{M}_6^1, \mathbf{M}_7^1\}$, then $GWN$ and $SN_1$ will believe the legitimacy of $A$ and they will build a shared session key successfully. The following procedures are similar to original scheme.
- Time Complexity: near to a legitimate user.
- The Scale of the Attack: (a) $\rightarrow$ (b) $\rightarrow$ (c) $\rightarrow$ (f) as shown in Fig. 4. In the beginning, the adversary $A$ gets $SN_1$’s private key (marked in red circle as shown in (a) of Fig. 4), then exploits messages among $GWN$, $SN_1$ and $U_1$, $A$ can compute session keys between $U_1$ and $SN_1$ as above. Since $A$ also gets all $SN$’s private key in the first attack, $A$ then can compute session keys between $U$ and $SN_1$ as shown in (i) of Fig. 4. When the number of the attacks is large enough, $A$ can compute previous session keys between all users and all sensor nodes as shown in (d) of Fig. 4.

From the above attacks, we can see that securely distributing the private key of the sensor nodes is fundamental to the whole security, it also decides the authentication process. Inappropriate private key distribution can cause sensor node impersonation and session key leakage.

4.3 Node Capture Attack Type III
Type III utilizes the vulnerability of insecure transmission of user’s fixed unique secret parameter $FUSP_{G/US}$ to obtain the necessary information to impersonate the user. This section introduces Type III via Li et al.’s scheme [34] as follows:

- Adversary’s Capability:
  (1) “C1”. Eavesdrop the authentication message between $GWN$ and $SN_1$ to get $M_6$ and $M_9$, and the message between $GWN$ and $U_1$ to get $M_{14}$ in the session of $SN_1$.
  (2) “C6”. Get $SN_1$’s secret key $K_{GWN-S}$.
- Attack Target: the users.
- Attack Consequence: get $FUSP_{G/US}$ to impersonate $U_1$.
- Attack Steps:
Step 1. Compute $ID_i = M_6 \oplus K_{GWN-S}$.
Step 2. Compute $r_5 = h(ID_i||K_{GWN-S} \oplus M_9)$.
Step 3. Compute $M_1 = M_{14} \oplus r_5$, once acquires $M_1$, $A$ has the ability to forge the message sent by $U_1$ to spoof $GWN$ and sensor nodes as follows:
Step 4. Generate $r_4$ and $S^4$.
Step 5. Compute $M_2^4 = S^4P$.
Step 6. Compute $M_3^4 = S^4X$, note that $X$ is a public parameter and can be easily gotten.
Step 7. Compute $M_4^4 = ID_i \oplus M_3^4$.
Step 8. Compute $M_6^4 = M_1^4 \oplus r_4^4$.
Step 9. Compute $M_7^4 = h(ID_i||r_4^4) \oplus \mathbf{SID}_m$ (m can be any valid number).
Step 10. Compute $M_8^4 = h(M_1^4||r_4^4)\oplus \mathbf{SID}_m$ (m can be any valid number).
Step 11. Send $\{M_2^4, M_4^4, M_6^4, M_7^4\}$, then $GWN$ and $SN_1$ will believe the legitimacy of $A$ and they will build a shared session key successfully. The following procedures are similar to original scheme.
- Time Complexity: near to a legitimate user.
- The Scale of the Attack: (a) $\rightarrow$ (b) $\rightarrow$ (c) $\rightarrow$ (f) as shown in Fig. 4. In the beginning, the adversary $A$ gets $SN_1$’s private key (marked in red circle as shown in (a) of Fig. 4), then exploits messages among $GWN$, $SN_1$ and $U_1$, $A$ can compute session keys between $U_1$ and $SN_1$ as above. Since $A$ also gets all $SN$’s private key in the first attack, $A$ then can compute session keys between $U$ and $SN_1$ as shown in (i) of Fig. 4. When the number of the attacks is large enough, $A$ can compute previous session keys between all users and all sensor nodes as shown in (d) of Fig. 4.

4.4 Node Capture Attack Type IV
Type IV is an complete impersonation where $A$ with the private key of $SN_1$ can get victim’s password and identity via offline dictionary attacks. Generally, offline dictionary attacks occur when $A$ can find a verification parameter VP to test the correctness of guessed value and is one of the most common attacks in user authentication schemes. Though as we mentioned in Sec. 1, Wang et al. [52] introduce the public-key technique to resist against offline dictionary attacks, the situation in WSNs is a little bit different where $A$ can obtain some sensor nodes’ private key to gain many advantages to conduct such an attack. Furthermore, when analyzing offline dictionary attacks, most protocol designers focus on the first two message flows, while pay little attention to subsequent flows sent by the gateway. This well-explains
why such an attack is ignored by Jiang et al. [30] when they already have known the way to apply public-key algorithm to withstand this attack.3

– Adversary’s Capability:
  (1) “C1”. Intercept the message between GWN and SNi to get M3, M7, T2 in authentication phase.
  (2) “C2”. Offline enumerate all items in the space of password and identity.
  (3) “C3”. Obtain biometrics fng and f1 in smart card.
  (4) “C6”. Get SNi’s private key Xi.
– Attack Target: the users.
– Attack Consequence: compute the password of Ui, then further impersonate Ui.
– Attack Steps:
  Step 1. Guess PWi to be PWi∗ and IDi to be IDi∗.
  Step 2. Compute K∗i = M5 ⊕ h(ID)|IDjXj|T2j.
  Step 3. Compute K′i = M7 ⊕ K∗i.
  Step 4. Compute SK = h(ID)|IDjK′j|K′j.
  Step 5. Compute Bj′ = BH(f, fng).
  Step 6. Compute di = f1 ⊕ h(ID)|PWj|Bj′.
  Step 7. Compute M2j = h(SK)|IDj|di|K′j.
  Step 8. Verify PWj′ and IDj′ by checking if M2j = M0.
  Step 9. Repeat Step 1 – 8 until the correct value of PWi and IDi are found.
– Time Complexity: O(|DPW| + |DID| + (4T H + T B)), where T B is the time for biometric-specific operation.
– The Scale of the Attack: (a)→(b)→(e)→(f) as shown in Fig. 4. The attack evolution is similar to that of Type III.

4.5 Node Capture Attack Type V

Type V contains two kinds basic attacks: (1) track users, or (2) compute users’ identity. Normally, there are three conditions to led to this attack. 1) A with SNi’s private key may exploit protocol’s unreasonable design intent to get users’ identity, such as Li et al.’s scheme [34] where SN is designed to get ID. 2) A also may make use of the vulnerability in a temporary-certificate-based scheme to track users, such as Wu et al.’s scheme [37], where A can compute TIDnew via eavesdropped D10, T4 and computed f0. Once with TIDnew, A is able to track Ui’s next message to learn users’ habits and preferences for business purpose. A limitation in the attack on Wu et al.’s scheme is that A can only trace activities of Ui just after Ui interacts with SNi. 3) A can exploit the insecure transmission of user identity to compute victim’s identity, and we take Amin et al.’s scheme [38] to explain this attack as follows:

– Adversary’s Capability:
  (1) “C1”. Eavesdrop the message between GWN and SNi to get {H, S1i, Vj, T2j} and K0j, and the message between GWN and Ui to get M2j during the authentication phase.
  (2) “C6”. Get SNi’s private key fj.
– Attack Target: the users.
– Attack Consequence: compute Ui’s identity IDi.
– Attack Steps:
  Step 1. Compute h(IDi) = SSj ⊕ h(fj|T2j).
  Step 2. Compute Kj = Vj ⊕ h(IDi).
  Step 3. Compute K′j = Kj ⊕ K0j.
  Step 4. Compute SK = h(h(IDj)||SIDj||K′j||K′j).
  Step 5. Compute IDj = M2j ⊕ h(SK||K0j).
– The Scale of the Attack: (a)→(b)→(e) as shown in Fig. 4. In the beginning, the adversary A gets SNi’s private key (marked in red circle as shown in (a) of Fig. 4), then exploits messages among GWN, SNi and Ui, A can compute Ui’s identity as above shown in (b) of Fig. 4. Similarly, A can get Uj’s identity. When the number of the attacks is large enough, A can get all users’ identity as shown in (e) of Fig. 4.

4.6 Node Capture Attack Type VI

Type VI depicts an attack where the adversary with SNi’s private key XSN can acquire SMn’s (m ≠ i). Two situations will result in this attack: (1) The inappropriate distribution of SNi’s private key. (2) The insecure transmission of SNi’s private key. We take Fan et al.’s scheme to show the first situation. In Fan et al.’s scheme, since all the sensor nodes share a same private key, once A compromises SN to get SK, then the private key of SMn is exposed too. Then A can impersonate all sensor nodes with SK.

We take Dhillon et al.’s scheme [41] as an example to show the second situation where A can learn SMn’s private key due to the insecure transmission of the key:

– Adversary’s Capability:
  (1) “C1”. Eavesdrop the message between GWN and SMn to get An, e1 and TS2, and the message between Ui and GWN to get TS1 in the session of SMn (m ≠ i).
  (2) “C6”. Get SNi’s private key XSN, note that XSN is a shared secret between GWN and SN.
– Attack Target: sensor node.
– Attack Consequence: get SMn’s private key.
– Attack Steps:
  Step 1. Compute ym = An + H(XSN||TS1||TS2).
  Step 2. Compute xnm = ym + em, note that xnm is a private key for SMn, thus now A can impersonate SMn. Since the interaction processes are the same as original scheme, we omit here.
– Time Complexity: O(4T H) for getting SMn’s private key.
– The Scale of the Attack: (a)→(i)→(d) as shown in Fig. 4. In the beginning, the adversary A compromises the sensor node SN1 (marked in red circle as shown in (a) of Fig. 4), then exploits messages among SN2, U and GWN to get the private key of SN2 as above. With SN2’s private key , A can impersonate SN2 to all users, as shown in (i) of Fig. 4. After enough interactions, A can get all sensor nodes’ private keys and impersonate any sensor nodes to any users, as shown in (d) of Fig. 4.

4.7 Node Capture Attack Type VII

Type VII presents an impersonation attack where the adversary with SNi’s private key gets U’s unique secret parameter, and then exploits GWN’s inefficient authentication to
SN to impersonate SN_m (m ̸= j). It usually occurs in the communication model (b) of Fig. 5. Type VI and Type VII both are about impersonating sensor node, while Type VI is the complete impersonation, Type VII is an elaborate camouflage, i.e., “Incomplete Impersonation”. We take Kumari et al.’s scheme [42] as an example to explain this attack:

- **Adversary’s Capability:***
  1. “C1”. Eavesdrop D_g^i in the session between GWN and SN_i. Furthermore, A joins the session actively, intercepts and modifies messages among participants.
  2. “C6”. Get SN_j’s private key TC_j.
- **Attack Target:*** sensor node.
- **Attack Consequence:*** get U_i’s unique secret parameters, then impersonate SN_m (m ̸= j) to U_i.
- **Attack Steps:***
  Step 1. Compute h(ID_q)||h(Q_i)) = D_g^i ⊕ h(TC_j).
  Step 2. Intercept message to SN_m: {D_g^m, D_g^m, C_m^m, T_m^m}, note that this session is among U_i, SN_m and GWN.
  Step 3. Compute I_3 = D_g^m ⊕ D_g^m ⊕ h(ID_q)||h(Q_i)).
  Step 4. Select a random number K_3^i, compute S_3^i = T_k^i[h(ID_q)||h(Q_i))] mod p.
  Step 6. Compute S_3^i = h(SK_3^i)h[ID_q||h(Q_i))] [T_3^i], where T_3^i is timestamp.
  Step 7. Send {S_3^i, S_3^i, T_3^i} to U_i, then according to the protocol, U_i will authenticate A successfully and share session key SK_3^i with A.
- **Time Complexity:*** close to legitimate SN_m.

- **The Scale of the Attack (a)→(b)→(c)→(d) as shown in Fig. 4.** In the beginning, A compromises the sensor node SN_1 (marked in red circle as shown in (a) of Fig. 4), then exploits messages among U_1, SN_1 and GWN to get U_1’s FUSP_G/W (marked in red circle as shown in (b) of Fig. 4) as above. Similarly, A can get U_2’s FUSP_G/W. When the number of the attacks is large enough, A can impersonate any sensor nodes to any users, as shown in (g) of Fig. 4.

4.8 Node Capture Attack Type VIII

Type VIII usually happens where user’s login request first sends to a sensor node rather than the gateway such as model (c) of Fig 5. In this case, once the request is not well marked the target sensor node, A can intercept it and then carry out an impersonation attack. The similar attack can be found in Shi et al.’s scheme [53] criticized by Choi et al. [36]. In this section, we introduce node capture attack Type VIII via Farash et al.’s scheme [45].

- **Adversary’s Capability:***
  1. “C1”. Intercept messages and send messages to GWN.
  2. “C6”. Control SN_i, i.e., A gets x_i, h(X_GWN)||1 and joins the communication among U_i, SN_m and GWN actively.
  - **Attack Target:** sensor node.
  - **Attack Consequence:** impersonate SN_m.
  - **Attack Steps:***
    Step 1. Intercept {M_1, M_2, M_3, T_1} from U_i to SN_m, note that this session is among U_i, SN_m and GWN.
    Step 2. Compute ESID_4 = ESID_4 h(hX_GWN)||1] [T_2^i].
    Step 3. Select K_4^i, compute M_2^i = h(x_i)[T_1^i][T_2^i] + K_4^i.
    M_2^i = h(SID_4)[M_1^i][T_2^i][K_4^i].
    Step 4. Send GWN {M_1, M_2, M_3, T_1, T_2, ESID_4, M_4^i, M_5^i}.
    Step 5. GWN responds {M_6, M_7, M_8, M_9, T_3} to SN_j (i.e., the adversary), A computes K_3^i = M_7 + h(x_i)|T_3|.
    SK_3^i = (K_3^i + K_4^i), M_4^i = h(SK_4^i)[M_6][M_8][M_9][T_3^i].
    sends {M_6, M_8, M_9, T_3, T_4} to U_i.
    Step 6. U_i will authenticate A successfully, that is, U_i thinks that she shares the session key with SN_m, while actually with SN_j (i.e., the adversary).
- **Time Complexity:** close to a legitimate sensor node.

4.9 Node Capture Attack Type IX

In Type IX, the adversary A exploits the insecure transmission or distribution of GWN’s long term secret key to obtain the secret key. In this section, we show the details of Type IX via analysis of Das et al.’s scheme [49]:

- **Adversary’s Capability:**
  1. “C6”. Get SN_j’s private key MK.CH.
  2. “C7”. Register as a legitimate user U_i.
  - **Attack Target:** the gateway.
  - **Attack Consequence:** get GWN’s long-term secret key.
  - **Attack Steps:** note that K_1 = ESID_CH[ID_q][ID_GWN][X_s] is a special parameter related to GWN and SN_j and stored in U_i’s smart card, where X_s is GWN’s long-term secret key and MK.CH is SN_j’s private key. Once A collude with U_i or register as a legitimate user U_i and get K_1 from U_i’s smart card, then A can obtain X_s via decrypting K_1 with MK.CH.
- **Time Complexity:** O(T_S), where T_S is the operation time for symmetric encryption and decryption.

- **The Scale of the Attack:** (a)→(h) as shown in Fig. 4. In the beginning, A compromises the sensor node SN_1 (marked in red circle as shown in (a) of Fig. 4), then exploits U_1’s smart card to get GWN’s long-term secret key as above. After the first attack, A can obtain all unique secret parameters of U and SN_j, because their secret parameters are computed via this secret key. Thus, all participants and sessions are affected, as shown in (h) of Fig. 4.

4.10 Node Capture Attack Type X

In Type X, the adversary A with X_SN can modify session key between U and SN_j (or SN_m) without being noticed by any participants. After the attack, U and SN_j (or SN_m) do
not share the same session key and $A$ can compute the same session key as $U$. But the authentication is finished successfully. Generally, this attack includes two situations:

1. $A$ can modify session keys between U and compromised sensor node SN$_i$ due to users’ ineffective verification to CP$_{SK/S}$. Taking Amin et al.’s scheme [50] as an example, the last message of this scheme can be modified by $A$ as $\{M^4_{g8}, M^4_{g9}, M^4_{t1}, M^4_{t1} \}$, where $M^4_{g8} = h(R^5_{f1} \oplus R^5_{f2}), M^4_{g9} = M^5_g \oplus R^5_{f2} \oplus R^4_{t}, SK^4 = h(M^5_g \| R^4_{t} \| R^5_{f2}), M^4_{t1} = h(ID_i \| SK^4 \| R^5_{f2}), M^4_{t1} = M^1_t \oplus h(R^5_{f2} \oplus R^5_{f2}) \oplus h(R^5_{f2} \oplus R^5_{f2})$, note that ID$_i$ can be view as a known value to $A$. $R^5_{f2} = M^5_g \oplus h(SK_{GW/SN_i})$ (SK$_{GW/SN_i}$ is SN$_i$’s private key) and $R^5_{f2} = M^5_g \oplus R^5_{f2}$. In this way, the authentication is finished successfully, yet $U_i$ and SN$_i$ do not share the same session key.

The scale of the attack is (a)−(c) as shown in Fig. 4. In the beginning, the adversary $A$ compromises sensor node SN$_1$ (marked in red circle as shown in (a)), and makes SN$_1$ and U share different session keys as above. Similarly, $A$ can then modify session keys between SN$_1$ and U$_2$. When the number of the attacks is large enough, $A$ can modify session keys between all users and SN$_1$ as shown in (c).

2. In addition to problems in the first case, if CP$_{SK/U}$ is transmitted insecurely too, the second situation occurs where $A$ can modify all session keys between U and SN to make participants cannot share the same session keys, though the authentication is finished successfully. We use Amin et al.’s scheme [38] to show the attack:

- Adversary’s Capability:
  
  (1) “C1”. Eavesdrop $SS_j$ and $T_2$ from message between SN$_1$ and GWN, intercept and send message to GWN.

  (2) “C6”. Get SN$_j$’s secret key $f_j$.

- Attack Consequence: modify session key without noticed ($U_i$ and SN$_m$ (m ≠ j) share a different session key), meanwhile the authentication is finished successfully, and $A$ can compute the same session key as $U_i$.

- Attack Steps:

  Step 1. Compute $h(ID_i) = SS_j \oplus h(f_j || T_2)$.

  Step 2. Eavesdrop $V_m$ from GWN to SN$_m$ (m ≠ j), compute $K_j = V_m \oplus h(ID_i)$.

  Step 3. Intercept $\{M_1, K_m, T_4\}$ from GWN to $U_i$.

  Step 4. Generate $K^4_m$ which has the same length as $K_m$.

  Step 5. Compute $SK^4_m = h(h(ID_i) \| S{ID_m} \| K_j \| K^4_m)$.

  Step 6. Compute $K^4_1 = K^4_m \oplus K_m$.

  Step 7. Compute $M_1^4 = h(SK^4_m \| K^4_1 \| T_4)$.

  Step 8. Send $\{M_1^4, K^4_1, T_4\}$ to $U_i$, then $U_i$ will authenticate the message successfully.

  Step 9. Intercept $\{M_2\}$ from $U_i$ to GWN.

  Step 10. Compute $K_m = K^4_m \oplus K_j$.

  Step 11. Compute $M_2^4 = M_2 \oplus h(h(ID_i) \| S{ID_m} \| K_j \| K^4_m) \| K_j \| h(SK^4_m \| K_j)$.

  Step 12. Send $\{M_2^4\}$ to GWN. Finally, $U_i$ and SN$_m$ do not share the same session key, yet $U_i$ and $A$ do.

- Time Complexity: $O(6T_1)$.

5 Suggestions to Node Capture Attacks

Sensor nodes are usually deployed in unattended environments, thus it is easy for an adversary to breach some sensor nodes and extract the data stored in them. Based on this reality, it is very important to ensure the security of the system after the sensor node is compromised. Much effort has been taken to design a secure scheme resisting such an attack, while most attempts failed. In this section, we summarize the rationales for node capture attacks, put forward some suggestions to avoid such attacks.

From attack consequences of Table 4, the adversary can 1) compute session keys from Type I and II, 2) impersonate users from of Type III and IV, 3) avoid user anomy from Type V, 4) impersonate sensor nodes from Type VI, VII and VIII, 5) get the long-term secret key from Type IX, 6) modify session keys from Type X. Based on these six consequences, we figure out their causes from the vulnerabilities exploited of Table 4. With the six consequences and their causes, we draw the fishbone of node capture attacks, as shown in Fig. 6. It clarifies the causes of node capture attacks.

Fig. 6. The fishbone of node capture attacks.

- The Scale of the Attack: (a)→(b)→(e)→(f) as shown in Fig. 4. In the beginning, the adversary $A$ compromises SN$_1$ (marked in red circle as shown in (a) of Fig. 4) and gets some parameters of $U_1$ to modify session key between $U_1$ and SN$_1$ as above as shown in (b) of Fig. 4. Similarly, $A$ can get $U_2$’s $h(ID_2)$ and modify session key between $U_2$ and SN$_1$. With the increase in number of the attacks, $A$ can get all users’ useful parameters and modify session keys between all users and SN$_1$. When the number of the attacks is large enough, $A$ finally can modify session keys between all users and all sensor nodes as shown in (f) of Fig. 4.

Note that, in the first situation, the adversary $A$ can only modify session keys between compromised sensor node SN$_1$ and all users. In the second situation, $A$ finally can modify session keys between all sensor nodes and all users.
unreasonable design intent, insecure communication architecture, inappropriate distribution of SN’s private key, insecure transmission of FUSP\textsubscript{G/U} or some parameters, inefficient verification of SN or CP\textsubscript{SK/SN}, the issue of forward secrecy, the issue of offline dictionary attacks and the issue of temporary certificate. Suggestions to these eight issues are as follows:

- **Unreasonable Design Intent.** Note that user identity, unique secret parameter (FUSP\textsubscript{G/U} and TUSP\textsubscript{G/U}) cannot be known to sensor nodes, so do not let the gateway send these parameters to sensor nodes.

- **Insecure Communication Architecture.** From the viewpoint of node capture attacks, model (b) and model (c) of Fig. 5 are insecure: model (b) is likely to result in node capture attack Type VII as shown in Section 4.7. Model (c) is likely to result in node capture attack Type VIII as shown in Section 4.8. Furthermore, both model (b) and model (c) is bound to lead to node capture attack Type X. Following Wang et al.’s research [6], model (a) of Fig. 5 is recommended.

- **Inappropriate Distribution of SN’s Private Key.** The distribution of SN’s private key is a basic factor to the security of the system. Looking back Fan et al.’s scheme [29], it is very dangerous to have all sensor nodes share a same secret key. We recommend that let h(ID\textsubscript{SN}||x) be SN’s private key, this method has been accepted by most schemes [28], [30], [34], [54], [55]. In some of schemes, such as Gope et al.’s [56], GWN assigns a random unique secret number to SN as their private key. In this way, GWN must store the parameters related to the private keys of SN, which consumes more resources. Thus this method is not recommended.

Following this principle, the private key of sensor nodes in Fan et al.‘s scheme [29] should be h(ID\textsubscript{SN}||x) rather than a common shared key.

- **Insecure Transmission of Some Parameters, including the private key X\textsubscript{SN} of SN, unique secret parameters/identity ID of users, and long-term secret key.**

  The ways that the unique secret parameters are transmitted are varied from one protocol design to another. It is difficult to generalize, some basic principles are as follows:

  - Transmitting these parameters with “XOR” or symmetric encryption operation is dangerous, see Sections 4.3, 4.5, 4.6 and 4.9. We recommend to protect these parameters (denoted as Impor\textsubscript{par}) in a form of h(Impor\textsubscript{par}||x), where x denotes any parameters. Particularly, when Impor\textsubscript{par} is FUSP\textsubscript{G/U}, “x” has to include TUSP\textsubscript{G/U} (it is constructed by a public-key technique).

  Following this principle, we can fix Li et al.‘s scheme [34] by setting M\textsubscript{5} = h(M\textsubscript{1}||M\textsubscript{3} || f\textsubscript{1}) and M\textsubscript{14} = h(M\textsubscript{3}||M\textsubscript{1} || f\textsubscript{2} and all ID, in the parameters that GWN sends to SN be replaced with h(ID||K\textsubscript{GWN-S}), where M\textsubscript{1} is FUSP\textsubscript{G/U} and M\textsubscript{3} is TUSP\textsubscript{G/U}. In this way, the adversary cannot follow the steps in Section 4.3 to extract FUSP\textsubscript{G/U}, so this scheme can resist against the attack Type III, IV, V and X.

  - In some occasions, ID and X\textsubscript{SN} need to be transmitted with the operation “XOR”, we recommend to transmit them in the form of ID ⊕ h(TUSP\textsubscript{G/U}||x) and X\textsubscript{SN} ⊕ h(TUSP\textsubscript{G/SN} ||x) (or X\textsubscript{SN} ⊕ TUSP\textsubscript{G/SN} ||x), respectively. Note that TUSP\textsubscript{G/U} is constructed by a public-key technique.

  Following this principle, we can fix Li et al.’s scheme [57] by setting D\textsubscript{IDU} = ID\textsubscript{U} ⊕ h(D\textsubscript{1}|D\textsubscript{2}), D\textsubscript{3} = SN\textsubscript{A} ⊕ h(B\textsubscript{2}|D\textsubscript{2}), where B\textsubscript{2} is FUSP\textsubscript{G/U} and D\textsubscript{2} is TUSP\textsubscript{G/U}. In this way, the adversary cannot extract FUSP\textsubscript{G/U}, so this improved scheme is secure against the attack Type III and IV.

  - **Inefficient Verification of SN or CP\textsubscript{SK/SN}.** It contains two aspects: 1) GWN fails to authenticate SN. 2) users fail to verify CP\textsubscript{SK/SN}. To the first aspect, the first thing is to use a proper communication model. Then, do not merely rely on X\textsubscript{SN} to finish the authentication between SN and GWN, a temporary challenge, such as R\textsubscript{G/S} in Fig. 3, is necessary. Specifically, let Auth\textsubscript{3} at least contain FUSP\textsubscript{G/SN}, TUSP\textsubscript{G/SN} and CP\textsubscript{SK/SN}. It can stop the adversary from forging messages to fool GWN. To the second aspect, a public-key technique is required to construct TUSP\textsubscript{G/U}, and Auth\textsubscript{4} should at least contain TUSP\textsubscript{G/U} and CP\textsubscript{SK/SN}. This method can stop the adversary from modifying session keys (see Section 4.10).

Following above principle, to improve Amin et al.’s scheme [38], we first introduce ECC-based public-key technique with a pair of private/public key (X\textsubscript{GWN}, X\textsubscript{GWN} : P = Y), where P is a point on an elliptic curve which is built over prime finite field F\textsubscript{p}. Next, we construct TUSP\textsubscript{G/U} as TM\textsubscript{2} = K\textsubscript{g}, and let U\textsubscript{ID} send TM\textsubscript{1} = K\textsubscript{t}, P to GWN. Till now, U\textsubscript{ID} and GWN share the TUSP\textsubscript{G/U} (GWN can obtain TM\textsubscript{2} by computing X\textsubscript{GWN} · TM\textsubscript{1}). Next, we set the Auth\textsubscript{3} of Amin et al.’s scheme [38] and set session keys SK = h(TM\textsubscript{1}||TM\textsubscript{2})||TM\textsubscript{1}. Note that TM\textsubscript{1} is not transmitted in any channel. In this way, Aim et al.’s scheme [38] can achieve forward secrecy and resist against the attack Type I and Type II.

Following this scheme, we continue to improve Aim et al.’s scheme [38]. We let SN\textsubscript{A} compute two point multiplication operations: TM\textsubscript{3} = K\textsubscript{t} · P and TM = K\textsubscript{e} · TM\textsubscript{1}, and set session keys SK = h(TM\textsubscript{1}||TM\textsubscript{2})||TM\textsubscript{1}. Note that TM\textsubscript{1} is not transmitted in any channel. In this way, Aim et al.’s scheme [38] can achieve forward secrecy and resist against the attack Type I and Type II.

- **Offline Dictionary Attacks.** To resist offline dictionary attacks, the public-key algorithm is indispensable [52]. Yet, there is a subtlety worth noting: when accessing offline dictionary attacks, in addition to focusing on VP in login request initiated by U, special attention should be taken to Auth\textsubscript{3} and parameters with FUSP\textsubscript{G/U} in the channel, and this is often ignored by the protocol designer. As we have shown in Section 4.4, the adversary can exploit Auth\textsubscript{4} (M\textsubscript{5})
to carry out a dictionary attack successfully in the Jiang et al.’s scheme [30]. A recommended solution is to use \( FUSP_{G,U} \) in a form of \( h(FUSP_{G,U}||TUSP_{G,U}||\ast) \), where \( \ast \) denotes any valid numbers, and \( TUSP_{G,U} \) is constructed by a public-key technique.

Following this principle, we can improve Jiang et al.’s scheme [30]. First, we need to construct a \( TUSP_{G,U} \). The way to construct such a \( TUSP_{G,U} \) has been introduced above, therefore, the details are omitted. Second, we let \( M_U = h(d_U||TUSP_{G,U}||ID_U||K_j) \). In this way, Jiang et al.’s scheme [30] is resistant to the attack Type V.

- The Issue of Temporary Certificate. Applying temporary certificate algorithm to multi-factor user authentication schemes leads many problems [52], how to avoid these problems is still an open question. A simple way is that do not use temporary certificate technique. Actually, many schemes that do not use temporary certificate technique [26], [27], [28], [57] achieve as least the same security as those using this technique [37], [50], [56].

To design a secure authentication scheme that is resistant to node capture attacks, the above eight challenges should be taken into account. The specific method to achieve the above suggestions may be different from scheme to scheme, but we summarize some common principles against node capture attacks: 1) Regarding “Insecure communication architecture”, model (b) and model (c) of Fig. 5 are not secure against node capture attacks, and model (a) is recommended. 2) Users’ identity and unique secret parameter should be kept anonymous to sensor nodes. 3) It is recommended to set \( h(IDSN_j||x) \) as SN’s private key. Furthermore, as shown in Appendix A, which can be found at https://bit.ly/2VjHqY1 and also on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2020.2974220, we take Li et al.’s scheme [57] as an example to show a viable way to follow the above principles to avoid node capture attacks.

6 A COMPARATIVE EVALUATION OF EXITING SCHEMES FOR WSNs

Based on our taxonomy of node capture attacks in Section 3, we naturally improve our evaluation criteria by expanding the criterion “resistance to node capture attacks” into ten sub-criteria. We then perform a large-scale assessment of 61 multi-factor user authentication schemes for WSNs under our expanded criteria set and our attack model in Table 5. The selected schemes usually represent a typical attack or have attracted much attention and lead many new enhanced versions. This comparison gives a fair and comprehensive evaluation of existing schemes. Unsurprisingly, two early schemes, which were proposed around the year of 2005 when Benenson et al. [18] for the first time introduce node capture attacks into remote user authentication, are worse than other schemes. As time goes by, the situation gets better, which is in line with our understanding on the development of things. From Table 5, it is easy to see that so far no scheme meets all evaluation criteria after nearly ten years of research. The scheme with the best performance proposed by Li et al. [58] can only achieve at most 20 criteria, highlighting the unsatisfactory situation of user authentication schemes for WSNs.

When dividing the evaluation criteria into two parts, i.e., ideal attributes and security requirements, we can see another trend in the development of user authentication schemes, that is, the schemes’ performance in ideal attributes gets better and better. The challenge in satisfying the requirements of ideal attributes is to design schemes without relying on clock synchronization. Compared with the realization of ideal attributes, meeting the security requirements is more difficult. Every criteria of security requirements except S1 are all difficult to meet. In most cases, offline dictionary attacks are the main kinds of attacks for S5, and it can be stopped by applying public-key techniques correctly [12], [52]. In WSNs, S4 “resistance to known attacks” is becoming difficult as the complexity of system increases. According to Li et al.’s recent study [80], the administrator of the gateway can exploit the user’s login request as a verifier to guess victims’ passwords. Most schemes are vulnerable to such an attack and thus cannot satisfy the criteria S2 “No Password Exposure”. S3 “forward secrecy” is a tricky problem in WSNs because of the recourse-limited sensor nodes. “How to efficiently achieve forward secrecy in user authentication scheme for WSNs” is still an open issue.

Among all criteria, S6 “resistance to node capture attacks” is the hardest criterion to be achieved. As shown in Table 5, only two schemes meet S6. However, this trend cannot be reflected well under other evaluation criteria sets. For example, the schemes of Li et al. (2018 JNCA) [34], Wu et al. (2017 PPNA) [37], Wang et al. (2018 Sensors) [28], Jiang et al. (2017 IEEE Access) [30] and Das et al. (2016 SCN) [65] are thought of being resistant to node capture attacks under Wang et al.’s criteria set [6] where node capture attacks are included in the criterion “resistance to known attacks”, but these schemes are demonstrated that they cannot resist against node capture attacks under our criteria set. All this highlights the urgency and significance of understanding the failure in node capture attacks and the difficulty in designing a user authentication scheme for WSNs resistant against node capture attacks. Furthermore, each sub-criterion of S6 is met or unmet by at least ten schemes. This indicates that each of the ten sub-criteria is necessary and our taxonomy of node capture attacks is reasonable. We present a detailed discussion on S6 in Appendix B, available at https://bit.ly/2VjHqY1.

7 CONCLUSION

In this paper, we have taken the first substantial step towards systematically exploring node capture attacks against user authentication protocols for WSNs. We first define the adversary model, and then develop a detailed and through evaluation criteria including the effects of node capture attacks. We then categorize node capture attacks into ten different types in terms of the attack targets, adversary’s capabilities and vulnerabilities exploited. Next, we elaborate on each type of attacks through examining 11 typical vulnerable protocols and investigate the corresponding countermeasures. Finally, we extend our evaluation criteria and conduct a large-scale comparative measurement of 61 representative user authentication schemes for WSNs. Among those schemes, only two are secure against node capture attacks, highlighting the difficulty in designing node-capture-attack resistant user authentication schemes for WSNs and demonstrating the significance of our systematic study on node capture attacks.


### TABLE 5

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Year</th>
<th>Ref</th>
<th>Evaluation Criteria</th>
<th>Computational Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al.</td>
<td>2019</td>
<td>[54]</td>
<td></td>
<td>$3T_T + 10T_H + T_B + T_B$</td>
</tr>
<tr>
<td>Gopala et al.</td>
<td>2019</td>
<td>[59]</td>
<td></td>
<td>$2T_E + T_H + T_N$</td>
</tr>
<tr>
<td>Li et al.</td>
<td>2018</td>
<td>[57]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Senivas et al.</td>
<td>2018</td>
<td>[60]</td>
<td></td>
<td>$5T_H + 7T_L$</td>
</tr>
<tr>
<td>Wang et al.</td>
<td>2018</td>
<td>[55]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Wu et al.</td>
<td>2018</td>
<td>[54]</td>
<td></td>
<td>$4T_H + 7T_L$</td>
</tr>
<tr>
<td>Amin et al.</td>
<td>2018</td>
<td>[60]</td>
<td></td>
<td>$5T_H + 7T_L$</td>
</tr>
<tr>
<td>Li et al.</td>
<td>2018</td>
<td>[54]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Ali et al.</td>
<td>2018</td>
<td>[59]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Chalis et al.</td>
<td>2018</td>
<td>[60]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Wang et al.</td>
<td>2017</td>
<td>[58]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Smirnov et al.</td>
<td>2017</td>
<td>[61]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Jiang et al.</td>
<td>2017</td>
<td>[53]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Mes et al.</td>
<td>2017</td>
<td>[62]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Wu et al.</td>
<td>2017</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Tai et al.</td>
<td>2017</td>
<td>[57]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Dhyan et al.</td>
<td>2017</td>
<td>[54]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Park et al.</td>
<td>2017</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Gopal et al.</td>
<td>2016</td>
<td>[55]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Sridhar et al.</td>
<td>2016</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Kaur et al.</td>
<td>2016</td>
<td>[54]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Amin et al.</td>
<td>2016</td>
<td>[53]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Farnh et al.</td>
<td>2016</td>
<td>[55]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Kumar et al.</td>
<td>2016</td>
<td>[52]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2016</td>
<td>[53]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2016</td>
<td>[54]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Chang et al.</td>
<td>2016</td>
<td>[55]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Jiang et al.</td>
<td>2016</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Che et al.</td>
<td>2016</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Turkenovic et al.</td>
<td>2015</td>
<td>[53]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Yoon and Kum</td>
<td>2015</td>
<td>[54]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Xu et al.</td>
<td>2015</td>
<td>[55]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Shi et al.</td>
<td>2015</td>
<td>[56]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2015</td>
<td>[57]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2015</td>
<td>[58]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Kumar et al.</td>
<td>2015</td>
<td>[59]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2015</td>
<td>[60]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Kumar et al.</td>
<td>2014</td>
<td>[57]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Fan et al.</td>
<td>2014</td>
<td>[58]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2014</td>
<td>[59]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2014</td>
<td>[60]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2014</td>
<td>[59]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2014</td>
<td>[60]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2013</td>
<td>[61]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2013</td>
<td>[62]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2012</td>
<td>[63]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2012</td>
<td>[64]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2011</td>
<td>[65]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2011</td>
<td>[66]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2010</td>
<td>[67]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2010</td>
<td>[68]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Das et al.</td>
<td>2010</td>
<td>[69]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>Vaidya et al.</td>
<td>2009</td>
<td>[60]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
<tr>
<td>He et al.</td>
<td>2009</td>
<td>[70]</td>
<td></td>
<td>$7T_H + 5T_L$</td>
</tr>
</tbody>
</table>

* The case 1 of their scheme; \( P_2 \) of their scheme.

Some schemes do not describe dynamic sensor node addition or smart card revocation phases directly, but they do support the two phases, and thus meet D2. \( T_E, T_Y, T_C, T_H, T_B \) denote the operation time for modular exponentiation, elliptic curve point multiplication, chebysev chaotic-map, fuzzy extracting biometric data, hash, and symmetric encryption, respectively. Some lightweight operations like XOR and \( \oplus \) are omitted.

\( 
\begin{align*}
| & \text{scheme can provide the corresponding attribute.} \quad & \text{scheme cannot provide the corresponding attribute.} \quad & \text{attribute is not applied to the scheme.}
\end{align*}
\)

For example, if the scheme does not create a session key after authentication, then it does not make sense to discuss session key related security; if the scheme does not use password, then it does not make sense to discuss offline dictionary guessing attacks; if the message sent to users is an “acknowledgment” containing no sensitive parameters, then it does not make sense to discuss whether the adversary can modify this message.

### ACKNOWLEDGMENT

The authors thank Dr. Yanhong Xu and the anonymous reviewers for their invaluable comments. This research is supported by the National Key Research and Development Plan of China under Grant No. 2018YFB0803605, and by the National Natural Science Foundation of China under Grant No.61802006, and by the National Research Foundation, Prime Minister’s Office, Singapore under its Strategic Capability Research Centres Funding Initiative and Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S).


Chenyu Wang is currently working toward the PhD degree in the Beijing University of Posts and Telecommunications, and a visiting student in Nanyang Technological University. She has received the “Cyber security scholarship” (China) and has published several papers. Her research interests include cryptographic protocols and software security.
Ding Wang received the PhD degree in information security from Peking University, in 2017. He is currently a professor with the College of Cyber Science, Nankai University, and also serves as the deputy director of the Tianjin Key Laboratory of Network and Data Security Technology. As the first author, he has published more than 40 papers at venues like ACM CCS, Usenix Security, NDSS, IEEE DSN, ESORICS, ACM ASIACCS, ACM TCPS, IEEE Transactions on Dependable and Secure, and IEEE Transactions on Information Forensics and Security. Seven of them are recognized as “ESI highly cited papers”. His PhD thesis receives the “ACM China Doctoral Dissertation Award” and “China Computer Federation (CCF) Outstanding Doctoral Dissertation Award”. He has been involved in the community as a TPC, AEC member or PC Chair for more than 50 international conferences, such as Usenix Security, ACSAC, ISC, CNS, SEC, ACISP, and SocialSec. His research interests focus on authentication and provable security.

Yi Tu received the bachelor’s degree from Nankai University, China, in 2016, and the master’s degree from the George Washington University, in USA, in 2018. Currently he is working toward the PhD degree in the School of Physical and Mathematical Sciences in Nanyang Technological University. His research interests include machine learning, software security and symmetric key cryptanalysis.

Guoai Xu is currently a professor with the Beijing University of Posts and Telecommunications. He is a member of cyberspace security association in China. He has published more than 50 papers at venues like WWW, FSE/ESEC, and TECS. His research interests include information security, cryptographic, and software security.

Huaxiong Wang received the PhD degree in mathematics from the University of Haifa, Israel, in 1996, and the PhD degree in computer science from the University of Wollongong, Australia, in 2001. He joined Nanyang Technological University in 2006, and is currently an associate professor with the Division of Mathematical Sciences. He is also an honorary fellow at Macquarie University, Australia. His research interests include cryptography, information security, coding theory, combinatorics, and theoretical computer science. He has been on the editorial board of three international journals: Designs, Codes and Cryptography (2006–2011), the Journal of Communications (JCM), and the Journal of Communications and Networks. He was the program co-chair of Ninth Australasian Conference on Information Security and Privacy (ACISP 04), in 2004 and Fourth International Conference on Cryptology and Network Security (CANS 05), in 2005, and has served in the program committee for more than 70 international conferences. He has received the inaugural Award of Best Research Contribution from the Computer Science Association of Australasia, in 2004.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.