Foisting and Stealing of Keys in Sensor Networks

Peng Wang and Chinya Ravishankar

Department of Computer Science and Engineering
University of California, Riverside
{wangpe, ravi@cs.ucr.edu}

Abstract. We consider cryptographic key establishment in sensor networks without PKI or trusted third parties, using pairwise trust relationships between intermediaries. We describe a novel attack called *key foisting* that defeats current schemes, compromising 90% of the path keys with only 10% of the sensors in the network seized. We then present a two-way path-key establishment scheme that resists foisting. It reduces the probability of successful key foisting to nearly zero even with 20% of sensors seized. Its overhead is affordable, and its resilience is excellent.

1 Introduction

Sensor networks are now used in a wide variety of applications. Their ubiquity in our environment is exemplified by the Internet of Things (IoT), seen [1,2] as a self-configuring global network infrastructure based on interoperable protocols, comprising physical and virtual nodes with identities, attributes, and intelligent interfaces, integrated into a network. Its nodes will participate in information, business, and social processes, interacting with themselves and the environment, and influencing the real world by actions, with or without human intervention.

1.1 Heterogeneity Complicates Security

In large networks, sensors will have widely different configurations and hardware and software capabilities. Sensors may belong to different administrative domains, with different policies and protocols. No single set of policies or protocols will work for all sensors. Public-Key Infrastructures (PKIs) [3] can be an effective solution, but not all nodes in such a network may support or subscribe to PKIs. It is unlikely that any single third party will be sufficiently trusted to mediate symmetric pairwise key establishment between all nodes.

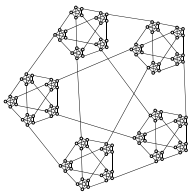
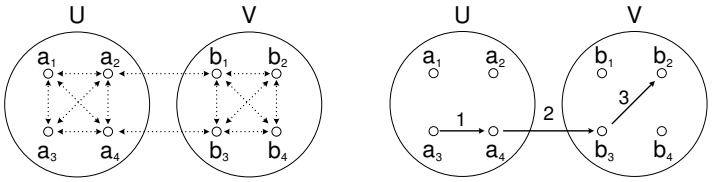


Fig. 1: Groups

Sensor Groups and Webs of Trust We note that it is natural to organize such large networks as groups, mirroring their structural, communication and trust relationships in the real world. Nodes in each organizational unit (floor, building, factory, vehicle, etc.) form a natural group for



(a) Associations between sensors

(b) Path key establishment

Fig. 2: Associations and path-key establishment. a_3 initiates path key establishment with b_2, via intermediaries a_4 and b_3, which are pairwise associated.

administrative purposes. Nodes in one group are more likely to communicate more with each other, and trust each other more than they might trust nodes from a different unit.

That last point is critical when neither PKI nor a globally trusted authority exists. We are then forced to exploit the trust relationships between nodes that naturally arise in a group-based organization. Such trust can be formalized as shared keys within and across groups at configuration time, and becomes the basis for the subsequent dynamic establishment of trust (keys) between nodes.

PGP [4] uses a similar "web of trust" model for decentralized public-key discovery. Users maintain validated user-public key associations in the form of personal "key rings". if user Alice needs Carol's public key, and Bob is able to forward a key he can cryptographically certify as Carol's, Alice can accept this key if she trusts Bob. When Alice only has partial trust in a set of users, she can accept a key if it is certified by a threshold number of users.

Assumptions and Threats Our work uses pair-wise trust relationships, rather than public keys. We first present key foisting, a new attack that easily compromises web-of-trust models, whether based on public or symmetric keys. We will then describe a two-way key establishment protocol that addresses this attack.

We show how to establish dynamic symmetric pairwise cryptographic keys when trusted authorities or PKI may be available to some, but not all nodes. Mutual trust must now be realized through pairwise shared symmetric keys between sensors. However, sensors lack enough memory to store all $O(n)$ pairwise keys for all other sensors in the system. Communication patterns are unknown in advance, so not all pairs of communicating sensors can share preloaded keys. We assume wireless sensor network environments, since they are inherently insecure [5–10]. However, our work applies equally to wired sensor networks.

Sensor Groups, Associations, and Agents Current group-based schemes [9, 11–14] try to establish shared keys without trusted third parties or PKI, but we will see that they have serious flaws. Typically, a WSN with n nodes is organized into g groups with γ nodes each [9]. Each node pair within a group U

is preloaded with a unique key. In Fig. 2a, each of $\{a_1, a_2, a_3, a_4\}$ holds a key for its neighbors in U. Also, $t > 1$ node pairs across each group pair (U, V) share preloaded keys. $(a_4, b_3), (a_2, b_1)$ are such pairs in Fig. 2a. Fig. 1 shows a 3-level group hierarchy. For a 2-level hierarchy with t agents across each group pair, each node holds only $\gamma - 1$ preloaded intra-group keys, and $t(g-1)/\gamma$ inter-group keys [9]. If $g = \gamma = \sqrt{n}$, each node holds only $O(\sqrt{n}) + O(t)$ keys, instead of the $O(n)$ pairwise keys required in a naive model.

Sensor pairs, such as (a_1, a_2) and (a_4, b_3) in Fig. 2a, that share preloaded keys are called *associated*. A sensor $s_i \in U$ sharing a key with a sensor $s_j \in V$ is an *agent* in U for V. Sensors not associated will establish *path keys* using agents as intermediaries. Fig. 2b shows typical path-key establishment in current schemes. a_3 establishes a path key with b_2, by forwarding it via agents a_4 and b_3. Hops $\langle a_3, a_4 \rangle$, $\langle a_4, b_3 \rangle$, and $\langle b_3, b_2 \rangle$ forward encrypted messages, possibly over multiple radio hops. Decryption and re-encryption occurs at a_4 and b_3.

By seizing a sensor, the adversary gains both its preloaded keys, as well as all path keys it mediates. Current schemes [9,11,14,15] recognize such attacks, which we call *key stealing* (**KS**). **KS** permits eavesdropping and false data injection. Typically, **KS** allows the adversary to compromise 30% of the path keys by seizing about 10% of its sensors [9, 14].

1.2 Our Contributions

We introduce *key foisting* (**KF**), a novel attack which can compromise 90% of the path keys by seizing only 10% of the sensors. There being no trusted third party, path key establishment must use trusted intermediaries, who can only authenticate on a hop-by-hop basis. *End-to-end authentication requires end-to-end keys, but the very purpose of path-key establishment is to set up such keys between the end points.* The adversary seizes intermediaries, fabricates path key establishment messages, and fools other sensors into accepting fake path keys.

Such attacks are devastating and hard to detect. In current schemes, such as [9, 11, 14], only about 1% of the communication channels are secured via preloaded keys. The rest are secured by path keys. The adversary can compromise 90% of path keys by seizing a mere 10% of sensors.

We address key foisting with a novel two-way (2W) scheme. This scheme is compatible with a variety of key management schemes. We apply our 2W scheme to mGKE [9] as an example, and present a rigorous analysis of resilience. Similar analysis is possible for other schemes.

Group-based schemes like [9] give sensors in the same group preloaded pairwise keys, and increase resilience to attacks and reduce overhead, since path keys between neighbors now requires local communication, unlike [15, 16]. Our two-way scheme can be combined with multipath reinforcement to resist hybrid attacks **KF-KS** (Section 6.1). Unlike [16] and [15], it is easy to find multiple disjoint key establishment paths between any two sensors in group-based schemes.

Related work appears in Sec. 2, and Sec. 3 presents an overview of key distribution, path key establishment, **KS** and **KF** attacks. Sec. 4 presents our two-way scheme and its use in mGKE. Sec. 5 analyzes one- and two-way path

key schemes. Sec. 6 analyzes replay attacks and hybrid **KF-KS** attacks against our two-way scheme. Sec. 8 concludes the paper.

2 Related Work

In [16], each sensor s_i randomly selects an m-subset S_i of a key pool K. Sensors s_i, s_j can use any key from $S_i \cap S_j$ as their shared key. If $S_i \cap S_j = \emptyset$, they can establish path keys via intermediaries. In the q-composition scheme [15], two sensors may set up a key if they share at least q preloaded keys. [15] generates an ID pool and a pairwise key pool for IDs. A sensor randomly selects an ID from the ID pool, and is preloaded with a key matching its ID from the key pool. Di Pietro et al. gave a rigorous analysis of security in random key predistribution schemes [17].

Threshold-based key predistribution is proposed in [12] and [13]. Blom's key space scheme [18] is improved in [12] using multiple key spaces. The polynomial-based key-predistribution scheme is expanded in [13] using a polynomial pool instead of a single polynomial. This scheme uses a logical grid in which all sensors on a row or columns share a key. Sensors on different rows or columns establish path keys via agents. Another grid-based scheme appears in [11]. GP [14] uses a grid, placing sensors on each row or column into the same group. Using unique pairwise keys always achieves the best resilience [9].

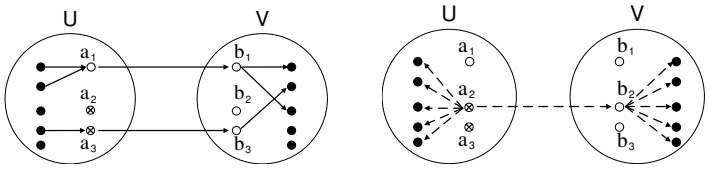
In [9], sensors in the same group share preloaded pairwise keys, and path keys established via agents are very robust. Intra-group keys have perfect resilience against key stealing. KeEs [6] guarantees backward and forward key security for key compromise attacks, but fails catastrophically for node compromises.

The potential of multipath reinforcement [15] is not realized by current schemes. They require disjoint cryptographic paths to be found on-demand, an expensive task. A cryptographic path may include many agents in [15], multiplying the chances of compromise. [19] guarantees paths with at most one agent, but requires flooding, which is too expensive. Fault localization is the focus in [20]. Other schemes [5, 10] try to mitigate the impact of false data injection attacks on in-network aggregation.

3 Attack Model

We illustrate our ideas using mGKE [9], which divides a sensor network into groups. All sensor pairs within each group share pairwise keys (i.e., are *associated*). For any two groups G_1, G_2, mGKE guarantees that at least one $s_i \in G_1$ and $s_j \in G_2$ share preloaded keys. Such sensors in different groups but sharing preloaded keys are called *agents*.

Fig. 2a shows two groups, each containing four sensors. All sensors within each group share preloaded pairwise keys. In Fig. 2a, (a_2, b_1) and (a_4, b_3) are the agent pairs between these groups. In Fig. 2b, sensors a_3 and b_2 establish a path key via agents a_4 and b_3. Let $K_{a_i b_j}$ be the key between nodes a_i and b_j, and $\langle M \| K_{a_i b_i} \rangle$ denote the message M encrypted with $K_{a_i b_i}$. To establish a path key



(a) Key Stealing (b) Key Foisting

Fig. 3: Key stealing and foisting. ⊗: seized agents.

with b_2, sensor a_3 picks a random value $K_{a_3b_2}$, and proceeds as follows (headers omitted for simplicity).

1. $a_3 \to a_4 : \langle (K_{a_3b_2}, a_3, b_2, G_v) \| K_{a_3a_4} \rangle$
2. $a_4 \to b_3 : \langle (K_{a_3b_2}, a_3, G_u, b_2) \| K_{a_4b_3} \rangle$
3. $b_3 \to b_2 : \langle (K_{a_3b_2}, a_3, G_u) \| K_{b_3b_2} \rangle$

Message (1), encrypted by a_3, may be relayed by several nodes before a_4 receives and decrypts it. Thus, $\langle a_3, a_4 \rangle$, $\langle a_4, b_3 \rangle$, $\langle b_3, b_2 \rangle$ are not radio hops, but encryption and decryption hops. A series of cryptographically active nodes mediating path keys, such as a_3, a_4, b_3, b_2 in Fig. 2b, is a *keypath*. The path key $K_{a_3b_2}$ is known to the end points a_3, b_2, but also to the agents a_4, b_3 that mediate the key. The adversary can get the key by seizing a_4 or b_3.

We assume the Yao-Dolev model [8,21]. The adversary may record all traffic, but wishes to remain undetected. Preloaded keys have perfect resilience [9], so we focus on threats to path keys. Cryptanalysis yields individual keys, but can be mitigated, as in [6]. We assume *node seizures*, a greater threat. Seizure yields all keys in a node, including path keys it mediates, and permits insider attacks [8].

In *key stealing* attacks, seized agents steal path keys they mediate [9, 11, 14, 15]. In Fig. 3a, agent a_3 is seized, and steals the path keys it mediates. a_2 is also seized, and can steal keys if used as agent. Keys mediated by a_1 are safe until it is seized. *Data injection* is a different attack, but also well-recognized [5, 10].

Redundancy can mitigate key stealing. A group pair G_u, G_v may have t agent pairs, each defining a keypath (Fig. 2a). A keypath is seized iff an agent within it is seized. Two sensors $s_i \in G_u$ and $s_j \in G_v$ can select any one of these t keypaths for key establishment, with probability $\frac{1}{t}$. An adversary who seizes c keypaths can seize this keypath with probability $\frac{c}{t}$. He succeeds with high probability only for high c, giving some protection against stealing.

3.1 Key Foisting: A Serious New Attack

The literature has not recognized that fraudulent path keys can be forced on victims by faked path-key establishment requests from seized agents. Such *key foisting* (**KF**) may be seen as an impersonation-and-key-injection attack. Injection has been studied for fake data [5, 10], but not path keys. Superficially similar, Sybil attacks [7, 22] overwhelm reputation systems with fake identities. Let $K_{s_xs_y}$ denote the preloaded key shared by some two nodes s_x and s_y.

1. Seize $s_a \in G_u$. Let s_a be an agent in G_u for groups $G_{v_1}, \ldots G_{v_k}$. Identify the pairs (s_i, s_j), $s_i \in G_u, s_j \in G_{v_l}, 1 \leq l \leq k$ served by s_a.
2. Target such a pair (s_i, s_j). Fabricate a key $K^*_{s_i s_j}$.
3. Fabricate a message that s_j wishes to establish key $K^*_{s_i s_j}$ with s_i, encrypt message with $K_{s_a s_i}$, and send to s_i. Now, s_i is tricked into accepting $K^*_{s_i s_j}$.
4. Let s_j belong to G_{v_l}, and let s_a be associated with agent $s_b \in G_{v_l}$. Fabricate a message claiming s_i wishes to establish key $K^*_{s_i s_j}$ with s_j. Encrypt this message with $K_{s_a s_b}$, and send it to s_b.
5. s_b accepts s_a's message, decrypts and re-encrypts it with $K_{s_b s_j}$, and forwards it to s_j, who is tricked into believing that the request originated with s_i.
6. s_i and s_j have been fooled into using $K^*_{s_j s_j}$.

Foisting defeats agent redundancy. *Seizing a single agent s_a suffices to foist fake path keys on all sensor pairs across all groups s_a serves.* In Fig. 3b, agent a_2 is seized, and sends fake path-key establishment requests to all sensor pairs it serves. Key foisting is feasible whenever path keys are established [11, 14], not just in group-based methods. All current schemes are vulnerable.

Foisting When Public Keys are Used Foisting is possible in public cryptography based key establishment schemes like PGP, when no certification authority is available. Let Alice and Bob have public keys P_A and P_B, and Carol have secret key S_C. Alice and Bob both trust Carol. They both know Carol's public key, but not each other's public key. The attack procceds as follows.

1. Seize Carol and her secret key S_C.
2. Fabricate two public keys P^*_A and P^*_B.
3. Claim that Alice wishes to establish a secure channel with Bob, and send the message $\langle (P^*_A) \| S_C \rangle$ to Bob. Now, Bob is tricked into accepting P^*_A since it is signed by S_C.
4. Claim that Bob wishes to establish a secure channel with Alice, and send $\langle (P^*_B) \| S_C \rangle$ to Alice, who accepts P^*_B, since it is signed with S_C.

Theorem 1 *Key stealing cannot be prevented if path keys are established using hop-by-hop intermediaries.*

Proof: If intermediaries $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ help establish path key K, this key is known to them all. The adversary can steal K by seizing any of these nodes. □

Key stealing cannot be prevented if PKI or a trusted authority is not available. We will present a scheme that prevents **KF** and strongly resists **KS**.

4 Two-Way Path-Key Establishment

We first briefly introduce mGKE [9]. mGKE preloads a unique key into each pair of sensors in the same group, so its intra-group resilience is perfect. In addition, t sensors pairs from $G_u \times G_v$ are preloaded with unique pairwise keys. Other sensors pairs use these agents to establish path keys (Fig. 2a). Each group

Notation	Meaning
n_s	number of sensors per group
n_g	number of groups
G_u	the uth group
$K_{s_i s_j}$	the pairwise key between s_i and s_j
t	number of agent pairs between groups
a^i_{uv}	the i-th agent in G_u for G_v
A_{uv}	the set of agents between G_u, G_v
$\langle M \| K_{s_i s_j} \rangle$	message M encrypted with key $K_{s_i s_j}$
$[K \backslash p]$	p-th share of a secret K
2W / 1W	Two-way/one-way path key establishment
k-PR	k-path key reinforcement
$K^{\rightarrow}_{s_i s_j}$	a half of the path key sent from s_i to s_j
$K^{\leftarrow}_{s_i s_j}$	a half of the path key sent from s_j to s_i

Table 1: Our Notation

contains n_s sensors, and there are n_g groups in the network. Agents in G_u for G_v are selected using the formula

$$F_{uv}(i) = (t(v-1) + i) \mod n_s, \tag{1}$$

where $F_{uv}(i)$ is ID of the i^{th} agent in G_u for G_v, t is the number of agents between groups. Let $A_{uv} = \{a^1_{uv}, \ldots a^t_{uv}\}$ denote this set of agents in G_u for G_v.

4.1 Two-Way Key Establishment (2W)

We propose two-way key establishment (2W) to deal with key foisting. All schemes to date have used one-way key establishment (1W). Let $s_j \in G_v$ receive the key-establishment request $\langle (K_{s_i s_j}, G_u, s_i) \| K_{s_j, a^1_{vu}} \rangle$ from an agent $a^1_{vu} \in G_v$, which is associated with $a^1_{uv} \in G_u$. In a 1W scheme, it is impossible for s_j to know whether $K_{s_i s_j}$ is legitimate or was faked by a compromised a^1_{uv} or a^1_{vu}.

In contrast, in our 2W scheme, s_i creates and sends a *forward half* $K^{\rightarrow}_{s_i s_j}$ of the path key to s_j, which responds with a *reverse half* $K^{\leftarrow}_{s_i s_j}$ via a disjoint path. s_i and s_j compute the path key as $K_{s_i s_j} = K^{\rightarrow}_{s_i s_j} \oplus K^{\leftarrow}_{s_i s_j}$. They can both trust $K_{s_i s_j}$ since each generated a part of it.

Forward Phase of 2W Key Establishment: $s_i \in G_u$ finds the agent set A_{uv} for G_v via Eqn. (1), and randomly selects an agent a^x_{uv}. Next, s_i encrypts a random $K^{\rightarrow}_{s_i s_j}$ with $K_{s_i, a^x_{uv}}$, sending it to a^x_{uv} in message $\langle (K^{\rightarrow}_{s_i s_j}, G_v, s_i, s_j) \| K_{s_i, a^x_{uv}} \rangle$. a^x_{uv} recovers $K^{\rightarrow}_{s_i s_j}$ and sends it encrypted to a^x_{vu} as $\langle K^{\rightarrow}_{s_i s_j}, G_u, s_i, s_j \| K_{a^x_{uv}, a^x_{vu}} \rangle$. s_j recovers $K^{\rightarrow}_{s_i s_j}$, and begins the reverse phase.

Reverse Phase of 2W Key Establishment: As shown in Fig. 2a, all keypaths between two sensors are disjoint. To ensure disjoint keypaths, s_j drops the agents

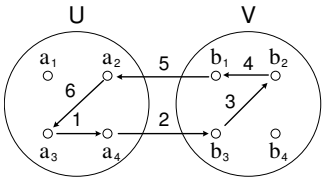


Fig. 4: 2W path key establishment in mGKE

used in the forward phase, and picks an agent a^y_{vu} from among the remaining $t-1$ agents in A_{vu}. s_j now picks a random values $K^{\leftarrow}_{s_i s_j}$ representing its half of the path key. s_j sends this half to s_i, exactly mirroring s_i's actions in the forward phase, but using the agents a^y_{vu} instead. The agent forwards $K^{\leftarrow}_{s_i s_j}$ to its peer agent in G_u, who forwards it to s_i. At the end of the reverse phase, s_i and s_j both have $K^{\rightarrow}_{s_i s_j}$ and $K^{\leftarrow}_{s_i s_j}$ and generate the path key $K_{s_i s_j} = K^{\rightarrow}_{s_i s_j} \oplus K^{\leftarrow}_{s_i s_j}$. We require $t \geq 2$.

Fig. 4 shows 2W path key establishment in mGKE. To establish a path key with b_2, sensor a_3 picks a random value $K^{\rightarrow}_{a_3 b_2}$, and proceeds as follows (message headers are omitted for simplicity).

1. $a_3 \rightarrow a_4 : \langle (K^{\rightarrow}_{a_3 b_2}, a_3, b_2, G_v) \| K_{a_3 a_4} \rangle$
2. $a_4 \rightarrow b_3 : \langle (K^{\rightarrow}_{a_3 b_2}, a_3, G_u, b_2) \| K_{a_4 b_3} \rangle$
3. $b_3 \rightarrow b_2 : \langle (K^{\rightarrow}_{a_3 b_2}, a_3, G_u) \| K_{b_3 b_2} \rangle$
4. $b_2 \rightarrow b_1 : \langle (K^{\leftarrow}_{a_3 b_2}, a_3, b_2, G_u) \| K_{b_1 b_2} \rangle$
5. $b_1 \rightarrow a_2 : \langle (K^{\leftarrow}_{a_3 b_2}, b_2, G_v, a_3) \| K_{a_2 b_1} \rangle$
6. $a_2 \rightarrow a_3 : \langle (K^{\leftarrow}_{a_3 b_2}, b_2, G_v) \| K_{a_2 a_3} \rangle$

4.2 *k*-Path Reinforcement (*k*-PR)

mGKE with our 2W scheme defeats foisting. However, as Theorem 1 shows, key stealing is always possible. Using k-path reinforcement [15] also adds resilience against key stealing. In k-path reinforcement [15], a key is cryptographically divided into shares, and sent along k node-disjoint paths to the destination, where it is reconstituted from the shares. The adversary must compromise all k of these paths to steal the key.

Randomized methods like RKP [15] only make probabilistic guarantees about network connectivity, without assuring that node degrees are at least k. Nodes of lower degree cannot use k-PR. Even when k disjoint paths exist, they are expensive to find. In contrast, k-PR works well in mGKE, where keypaths are all agent-disjoint (Fig. 2a), so it suffices to pick any k keypaths. In mGKE, keypaths have two or fewer agents, but paths in [15] may have any number of them. k-PR security drops as the number of agents per path grows in [15]. Finally, mGKE initiators can find agents from Eqn. (1), and send path key messages via standard routing, but initiators in [15] must themselves discover paths and select agents from them. Even worse, [19] uses broadcasting and flooding to find agents. Group-based schemes like mGKE have several desirable properties:

$\mathbf{K_{s_i s_j}}$	$\mathbf{K_{s_i s_j}}$ is path key between s_i, s_j
$\mathbf{K_{ij}^q}$	$\mathbf{K_{s_i s_j}}$ is type-q path key, $q = 1, 2, 3$
$\mathbf{c^{\langle ij \rangle}}$	c nodes are seized in the network as a whole
$\mathbf{b}$	b of the $2t$ agents between (G_u, G_v) seized
$\mathbf{b^{\langle ij \rangle}}$	same as $\mathbf{b}$, without s_i or s_j being seized.
$\widehat{\mathbf{K}}_{\mathbf{s_i s_j}}$	path key K_{ij} between s_i, s_j is stolen

Table 2: Notation

- $k \leq t$ agent-disjoint keypaths exist between any two nodes, as t exist between any two groups.
- $s_i \in G_u$ gets k agent-disjoint keypaths to $s_j \in G_v$ just by selecting k agents from the t agents in A_{uv}.
- Intra-group communication overhead is far smaller than inter-group communication overhead [9].

5 Analysis of Resilience

We start by analyzing the resilience of mGKE with 1W against **KF**. When a node is seized, all its keys are lost, including path keys it mediated between unseized nodes. The *resilience* of a path key scheme is hence judged [9] by *the rate at which keys between unseized sensors are lost, as sensors are seized.*

Let mGKE (1W k-PR) denote mGKE using 1W path key establishment and k-path key reinforcement, and mGKE (2W k-PR) denote mGKE using 2W path key establishment and k-path key reinforcement. In mGKE (2W k-PR), k keypaths are used by both the initiator and the recipient, so that $2k$ keypaths are used in all. In k-path reinforcement [15], a path key is divided into k shares. The initiator sends the k shares via k agent-disjoint keypaths. To steal the path key, the adversary must now seize an agent in each path.

Definition 1 *A set of keypaths $\{p_1 \ldots, p_k\}$ used in k-path reinforcement is a k-keypath.*

To compare the resilience against **KF** and **KS**, we first analyze the resilience of mGKE (1W k-PR) against both attacks. Three cases arise when K_{ij} is a path key for $s_i \in G_u, s_j \in G_v$:

1. neither s_i nor s_j is an agent for pair (G_u, G_v),
2. one of s_i or s_j is an agent for (G_u, G_v), or
3. both s_i and s_j are agents for (G_u, G_v).

Let $\mathbf{K_{s_i s_j}}$ denote the event that K_{ij} is a path key and let $\mathbf{K_{ij}^q}$ denote that K_{ij} is a path key matching case q above. Let $\mathbf{c^{\langle ij \rangle}}$ be the event that c nodes are seized in all, but neither s_i nor s_j is. Let $\mathbf{b}$ denote that b of $2t$ agents for (G_u, G_v) are

seized. Let $\widehat{\mathbf{K}}_{\mathbf{s_i s_j}}$ denote that the key K_{ij} between s_i and s_j is stolen. Let $\mathbf{b}^{\langle \mathbf{ij} \rangle}$ denote that b agents are seized, but not s_i or s_j. We are interested in determining

$$\Pr\left[\widehat{\mathbf{K}}_{\mathbf{s_i s_j}} \mid \mathbf{c}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}_{\mathbf{s_i s_j}}\right] = \sum_{q=1}^{3} \sum_{b=1}^{2t} \left(P_{\widehat{K}|b} P_{b|c} P_q \right) \tag{2}$$

where $P_{\widehat{K}|b} = \Pr[\widehat{\mathbf{K}}_{\mathbf{s_i s_j}} \mid \mathbf{b} \wedge \mathbf{c}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}^{\mathbf{q}}_{\mathbf{ij}}]$, $P_q = \Pr[\mathbf{K}^{\mathbf{q}}_{\mathbf{ij}}]$, and $P_{b|c} = \Pr[\mathbf{b} \mid \mathbf{c}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}^{\mathbf{q}}_{\mathbf{ij}}]$.

Each group pair has t agent pairs, and $n_s^2 - t$ path keys. Of these, $(n_s - t)^2$ path keys are of type-1, $2t(n_s - t)$ of type-2, and $t(t-1)$ of type-3. Clearly, $\Pr[\mathbf{K}^{\mathbf{1}}_{\mathbf{s_i s_j}}] = \frac{(n_s - t)^2}{n_s^2 - t}$, $\Pr[\mathbf{K}^{\mathbf{2}}_{\mathbf{s_i s_j}}] = \frac{2t(n_s - t)}{n_s^2 - t}$, and $\Pr[\mathbf{K}^{\mathbf{3}}_{\mathbf{s_i s_j}}] = \frac{t(t-1)}{n_s^2 - t}$. For simplicity, we only analyze the resilience of type-1 path keys. The resilience of type-2 and type-3 path keys can be easily obtained using the same method.

5.1 mGKE (1W k-PR) Key-Stealing Resilience

With Eqn. (2) in mind, the probability of event $\mathbf{b}$ with c seized sensors, excluding s_i and s_j is found as follows. For type-1 path keys, we can seize b of $2t$ agents in $\binom{2t}{b}$ ways, and seize $c - b$ sensors from n sensors, except for s_i, s_j and $2t$ agents, in $\binom{n-2-2t}{c-b}$ ways. Hence,

$$\Pr\left[\mathbf{b} \mid \mathbf{c}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}^{\mathbf{1}}_{\mathbf{s_i s_j}}\right] = \frac{\binom{n-2-2t}{c-b}\binom{2t}{b}}{\binom{n}{c}} \tag{3}$$

For type-1 path keys, if $b \leq k-1$, the adversary can steal no keypaths. If $b \geq 2t-1$, all keypaths are stolen. Define the ranges $R_1 = [0, k-1], R_2 = [k, 2(t-1)]$, and $R_3 = [2t-1, 2t]$. Now,

$$\Pr\left[\widehat{\mathbf{K}}_{\mathbf{s_i s_j}} \mid \mathbf{b}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}^{\mathbf{1}}_{\mathbf{s_i s_j}}\right] = \begin{cases} 0 & b \in R_1 \\ g_1 & b \in R_2 \\ 1 & b \in R_3 \end{cases} \tag{4}$$

$$g_1 = \sum_{l=\lceil \frac{b}{2} \rceil}^{\min(t,b)} \frac{\binom{t}{b-l}\binom{t-b+l}{2l-b}\binom{l}{k}}{\binom{2t}{b}\binom{t}{k}} \cdot 2^{2l-b} \tag{5}$$

In Eqn. (5), $\binom{l}{k}/\binom{t}{k}$ is the probability that the k-keypath used by s_i and s_j is seized, when l keypaths are seized. $\frac{\binom{t}{b-l}\binom{t-b+l}{2l-b}}{\binom{2t}{b}} \times 2^{2l-b}$ is the probability that l keypaths are seized, when b agents are seized. Eqns. (3–5) give the probabilities Eqn. (2) needs for type-1 path keys. Analysis for type-2 and type-3 keys is similar. Our analysis matches simulation, and resilience is excellent (Fig. 5). We simulate a sensor network with 10000 nodes under the mGKE (1W k-PR) scheme. Let c denote the number seized sensors, N_c denote the total number of path keys of $n - c$ unseized sensors and N_c^f denote the number of compromised

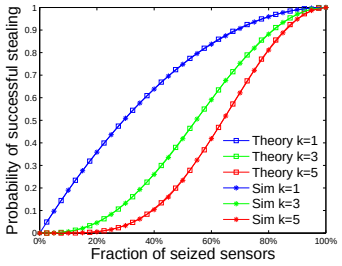


Fig. 5: mGKE (1W k-PR) key stealing resilience ($t = 10$, $n_s = n_g = 100$).

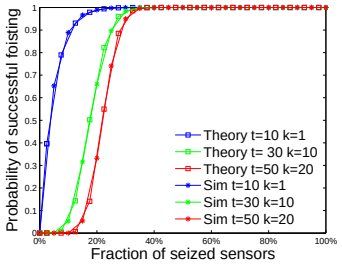


Fig. 6: mGKE (1W k-PR) foisting resilience ($n_s = n_g = 100$).

path key when c sensors are seized. We use the ratio $\frac{N_c^f}{N_c}$ as the probability of successful key stealing for various c. All simulations in this paper were conducted in this manner. With even 20% sensors seized, the chances that a given pathkey is stolen are under 5% for 3-PR, and under 1% for 5-PR. We see that k-path key reinforcement is very effective in dealing with **KS**. k-PR enhances the resilience to **KS** for methods other than mGKE as well [9, 11, 14]. However, we will now show that no 1W scheme can resist **KF**, despite the use of k-PR.

5.2 mGKE (1W k-PR) Key-Foisting Resilience

We now present the first-ever analysis of foisting. We show that all 1W schemes [9, 11, 14] perform poorly against **KF**. For type-1 path keys,

$$\Pr\left[\widehat{\mathbf{K}}_{\mathbf{s_i s_j}} \mid \mathbf{b}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K}^{\mathbf{1}}_{\mathbf{s_i s_j}}\right] = \begin{cases} 0 & 0 \le b \le k-1 \\ g_2 & k \le b \le 2t \end{cases} \tag{6}$$

$$g_2 = \sum_{l=\max(\lceil \frac{b}{2} \rceil, k)}^{\min(b,t)} 2^{2l-b} \frac{\binom{t}{b-l}\binom{t-b+l}{2l-b}}{\binom{2t}{b}} \tag{7}$$

In Eqns. (6, 7), all keypaths are secure when $b \le k-1$. Else, we can choose l agent pairs from $2t$ agents in $2^{2l-b}\binom{t}{b-l}\binom{t-b+l}{2l-b}/\binom{2t}{b}$ ways. Unlike Eqn. (4), $\frac{\binom{l}{k}}{\binom{t}{k}}$ is missing because all path keys can be foisted with any k keypaths seized.

We now have the probabilities needed in Eqn. (2). Eqns. (6) yield $P_{\widehat{K}|b}$, and Eqns. (3), (4) yield P_q and $P_{b|c}$. Fig. 6 shows the probability that a given path key in mGKE (1W) has been foisted, as per Eqns. (6–7) and simulation, for $k = 1, 10, 20$. Our analysis matches simulations perfectly. Comparing Fig. 5 and Fig. 6, it is clear that **KF** is much more devastating than **KS** and simply using multipath reinforcement cannot improve the resilience much.

An analysis of PIKE, GP to stealing is given in [9], but no analysis for foisting has appeared. Fig. 7 shows our simulation results of foisting resilience for PIKE-2D and GP (unique pairwise keys), with $n_g = n_s = 100$, $t = 10$. PIKE's good

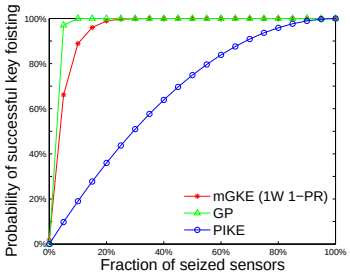


Fig. 7: Key-foisting resilience: mGKE (1W 1-PR), PIKE, GP.

showing is meaningless, given its poor resilience to stealing [9]. GP performs the worst, as its groups share too many agents.

5.3 mGKE (2W 1-PR) Resilience to Foisting

We analyze mGKE (2W 1-PR) performance, based on mGKE (1W). The following simple lemma is useful.

Lemma 1. *A keypath between G_u and G_v is compromised either in both directions, or not at all.*

Proof: A keypath is compromised iff one or more agents in it are. Agents are indifferent to message direction. ■

Theorem 2 *mGKE (2W k-PR) is immune to key foisting if $k > 0$, no matter how many nodes are seized.*

Proof: We assume that the adversary has knowledge of data local to any node *if and only if* he has seized the node. We yield him the maximum advantage, setting $k = 1$. Now, let him seize all sensors in G_u and G_v except s_i, s_j.

Assume the adversary foists a key $K^*_{s_i s_j}$ on s_i and s_j, so that neither s_i nor s_j functioned as initiator. By the 2W algorithm (Sec. 4.1), s_i must have received a share f^*_i from the adversary, generated a random g_i, and computed a key locally as $K^i_{s_i s_j} = f^*_i \oplus g_i$. Similarly, s_j must have computed $K^j_{s_i s_j} = f^*_j \oplus g_j$, using the locally generated random value g_j. Since the adversary foisted the key $K^*_{s_i s_j}$ successfully, $K^i_{s_i s_j} = K^j_{s_i s_j} = K^*_{s_i s_j}$.

Since the adversary knows $K^*_{s_i s_j}$, f^*_i and f^*_j, he can compute $g_i = K^*_{s_i s_j} \oplus f^*_i$ and $g_j = K^*_{s_i s_j} \oplus f^*_j$. However, g_i and g_j were randomly generated local values, which he can access only if he controls both s_i and s_j. This contradicts our assumption that he controls neither. ■

To make mGKE immune to **KF**, it suffices to use 2W path key establishment. Multipath key reinforcement is not required to guard against foisting. Other schemes, such as PIKE and GP can also adopt the 2W path key establishment to guard against **KF**.

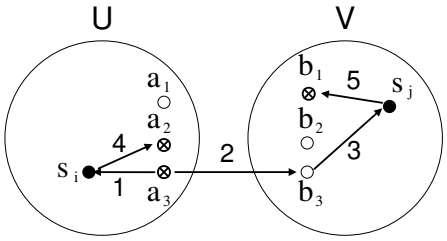


Fig. 8: **KF-KS** attack in mGKE (2W 1-PR)

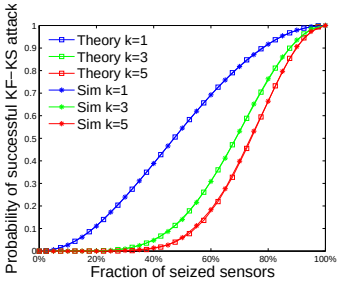


Fig. 9: mGKE (2W k-PR) resilience to man-in-the-middle attack, theory vs. simulation ($t = 10$, $n_s = n_g = 100$)

6 Replays, Key Foisting, and Man-in-the-Middle Attacks

Current 1W schemes do not guarantee message freshness, and are vulnerable to replays. Let a path key $K_{s_i s_j}$ established at time t_1 over k keypaths be compromised at time $t_2 > t_1$. In a 1W scheme, recording the inter-group path-key establishment messages at time t_1 allows the adversary to replay them at time $t_3 \geq t_2$, and foist $K_{s_i s_j}$ on s_j.

Theorem 3 *mGKE (2W) is immune to replays.*

Proof: Exactly as for Theorem 2. ■

Keys cannot be directly foisted in 2W schemes since the adversary cannot control the key half generated by receiver s_j. **KS** attacks remain viable (see Theorem 1), as is the following hybrid attack, when a very large number of nodes are compromised. mGKE (2W k-PR) continues to show excellent resilience.

6.1 Hybrid (KF-KS) Attacks

The adversary can combine **KF** with **KS** to compromise security, by creating separate keys with a pair of sensors and interposing himself in between. He must control enough agents in each group to control all key paths with high probability. We will show that **KF-KS** is no worse for mGKE (2W k-PR) than simple **KS**.

In Fig. 8, the adversary has seized agents $a_2, a_3 \in G_u$, $b_1 \in G_v$, and attacks $s_i \in G_u$ and $s_j \in G_v$ as follows.

1. Fabricate a forward half $K^{*\rightarrow}_{s_j s_i}$. Fabricate a message that s_j wishes to establish key with s_i. Encrypt message with $K_{a_3 s_i}$, and send to s_i. Now, s_i is tricked into accepting $K^{*\rightarrow}_{s_j s_i}$.
2. Fabricate another forward half $K^{*\rightarrow}_{s_i s_j}$. Fabricate a message that s_i wishes to establish key with s_j. Encrypt message with $K_{a_3 b_3}$, and send to b_3.
3. b_3 will forward the forward half to s_j. Now, s_j is tricked into accepting $K^{*\rightarrow}_{s_i s_j}$.

4. s_i generates its reverse half $K^{\leftarrow}_{s_j s_i}$ and sends it to s_2 via agent a_2. If the adversary has seized a_2, he can steal this reverse half, and also suppress the message. He now computes the path key $K^{*\rightarrow}_{s_j s_i} \oplus K^{\leftarrow}_{s_j s_i}$, which is used by s_i as the path key for s_j.
5. s_j generates its reverse half: $K^{\leftarrow}_{s_i s_j}$ and sends it to s_i via agent b_1. If the adversary has seized b_1, he can steal this reverse half and also suppress the message. He now computes the path key $K^{*\rightarrow}_{s_i s_j} \oplus K^{\leftarrow}_{s_i s_j}$, which is used by s_j as the path key for s_i.

The adversary can now mount a Man-in-the-Middle attack between s_i and s_j:

1. Send a false message $\langle (M_1) \| K^{*\rightarrow}_{s_i s_j} \oplus K^{\leftarrow}_{s_i s_j} \rangle$ to s_j. M_1 is encrypted with a key that s_j accepts, so s_j will accept this message.
2. When s_j responds to s_i with $\langle (M_2) \| K^{*\rightarrow}_{s_i s_j} \oplus K^{\leftarrow}_{s_i s_j} \rangle$, seize and suppress this message. Now decrypt the message, tamper with it, encrypt it using s_i's key $\langle (M_2') \| K^{*\rightarrow}_{s_j s_i} \oplus K^{\leftarrow}_{s_j s_i} \rangle$, and send the message to s_i. s_i will also accept the message.

We will now show that **KF-KS** *is no more effective against mGKE (2W k-PR) than a simple* ***KS*** *attack.* In mGKE (2W k-PR), k keypaths are used by both the initiator and the recipient, so that $2k$ keypaths are used in all. With l keypaths seized, the chances that all shares of r_i and r_j are stolen are $\left[\frac{\binom{l-k}{k}}{\binom{t-k}{k}}\right]^2$. Let $\widehat{\mathbf{M}}$ denote success of a **KF-KS** attack. For type-1 keys,

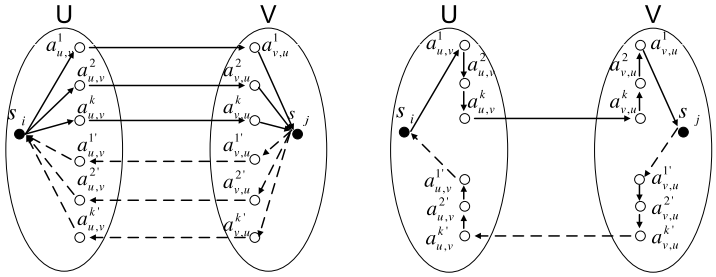
$$\Pr\left[\widehat{\mathbf{M}} \mid \mathbf{b}^{\langle \mathbf{ij} \rangle} \wedge \mathbf{K^1_{s_i s_j}}\right] = \begin{cases} 0 & b \in [0, 2k-1] \\ g_3 & b \in [2k, 2t] \end{cases}, \tag{8}$$

$$g_3 = \sum_{l=\max(2k, \lceil \frac{b}{2} \rceil)}^{\min(t,b)} 2^{2l-b} \frac{\binom{t}{l}\binom{l}{2l-b}}{\binom{2t}{b}} \left[\frac{\binom{l-k}{k}}{\binom{t-k}{k}}\right]^2. \tag{9}$$

Analysis of type-2 and 3 keys is similar. We get the resilience by using $P_{\widehat{M}|b}$ in Eqn. (8) to replace $P_{\widehat{K}|b}$ in Equation 2. Fig. 9 shows mGKE (2W k-PR)'s excellent resilience to **KF-KS** attacks, which succeed less than 12% of the time even with $k = 1$ and 20% of the sensors seized. mGKE (2W k-PR) has nearly perfect resilience against **KF-KS** even with $k \geq 3$ and 20% of sensors seized. It outperforms the original multipath reinforcement significantly (Fig. 6, 7). **KF-KS** is no more effective than **KS** (Fig. 5).

7 Message Aggregation To Reduce Overhead

The use of k-PR introduces additional messages, and hence additional overhead. We can reduce this overhead by a factor of 0.58 for $k = 2$, and 0.33 for $k = 5$, using the strategy shown in Figure 10. We omit the details of analysis due to lack of space. Aggregation is clearly effective. Surprisingly, 2WA can be more efficient than the very insecure 1W schemes. Our 2W method with aggregation provides both strong security and efficiency at the same time.



(a) Naive scheme. (b) Message aggregation.

Fig. 10: 2W path-key establishment

8 Conclusion

We have described key foisting, a new attack on sensor systems that has not so far been recognized in the literature, and showed how current schemes fail catastrophically against it. We then presented two-way key establishment which is practical in mGKE, and confers excellent resilience against foisting and related attacks, including man-in-the-middle attacks. We provided a detailed analysis of these attacks, and verified the accuracy of our analysis with detailed simulations. Our analysis and simulations confirm that mGKE (2W) has excellent resilience against both key stealing and foisting attacks. The two-way scheme has very low overhead compared even with the insecure one-way scheme. Our future work will include reducing the overheads even further, and implementing these schemes on real sensor networks.

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Comput. Netw. **54** (2010) 2787–2805
2. Vermesan, O., Harrison, M., Vogt, H., Kalaboukas, K., Tomasella, M., Wouters, K., Gusmeroli, S., Haller, S.: Internet of things—strategic research roadmap. Technical report, European Commission - Information Society and Media DG (2009)
3. Adams, C., Lloyd, S.: Understanding PKI: Concepts, Standards, and Deployment Considerations. 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
4. Schneier, B.: Applied cryptography: Protocols, algorithms, and source code in c, Wiley; 2nd endition (1995)
5. Chan, H., Perrig, A., Song, D.: Secure hierarchical in-network aggregation in sensor networks. In: CCS '06: Proceedings of the 13th ACM conference on Computer and communications security, New York, NY, USA, ACM (2006) 278–287
6. Di Pietro, R., Mancini, L., Jajodia, S.: Providing secrecy in key management protocols for large wireless sensors networks. Ad Hoc Networks **1** (2003) 455–468
7. Douceur, J.R.: The sybil attack. In: 1st International Workshop on Peer-to-Peer Systems. (2002)

8. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: First IEEE International Workshop on Sensor Network Protocols and Applications. (2002) 113–127
9. Ni, J., Zhou, L., , Ravishankar, C.V.: Dealing with random and selective attacks in wireless sensor systems. ACM Transactions on Sensor Networks **6** (2010)
10. Przydatek, B., Song, D., Perrig, A.: Sia: secure information aggregation in sensor networks. In: SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, New York, NY, USA, ACM (2003) 255–265
11. Chan, H., A.Perrig: Pike: Peer intermediaries for key establishment in sensor networks. In: Proceedings of IEEE Infocom. (2005) 524–535
12. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur. **8** (2005) 228–258
13. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, New York, NY, USA, ACM (2003) 52–61
14. Liu, D., Ning, P., Du, W.: Group-based key predistribution for wireless sensor networks. ACM Trans. Sen. Netw. **4** (2008) 1–30
15. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2003) 197
16. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, New York, NY, USA, ACM (2002) 41–47
17. Di Pietro, R., Mancini, L.V., Mei, A., Panconesi, A., Radhakrishnan, J.: Redoubtable sensor networks. ACM Trans. Inf. Syst. Secur. **11** (2008) 13:1–13:22
18. Blom, R.: An optimal class of symmetric key generation systems. In: Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, New York, NY, USA, Springer-Verlag New York, Inc. (1985) 335–338
19. Li, G., Ling, H., Znati, T.: Path key establishment using multiple secured paths in wireless sensor networks. In: CoNEXT '05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology, New York, NY, USA, ACM (2005) 43–49
20. Zhu, S., Setia, S., Jajodia, S.: LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. ACM Transactions on Sensor Networks (TOSN) **2** (2006) 528
21. Dolev, D., Yao, A.C.: On the security of public key protocols. Foundations of Computer Science, Annual IEEE Symposium on Foundations of Computer Science **0** (1981) 350–357
22. Newsome, J., Shi, E., Song, D., Perrig, A.: The sybil attack in sensor networks: analysis & defenses. In: IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks, New York, NY, USA, ACM (2004) 259–268