# SEDCOS : A Secure Device-to-Device Communication System for Disaster Scenarios

Kohnhäuser, Florian; Stute, Milan; Baumgärtner, Lars et al.

(2017)

# SEDCOS: A Secure Device-to-Device Communication System for Disaster Scenarios

F. Kohnhäuser*§, M. Stute*†, L. Baumgärtner‡, L. Almon†, S. Katzenbeisser§, M. Hollick†, B. Freisleben‡
§Security Engineering Group, TU Darmstadt, Germany, {kohnhaeuser,katzenbeisser}@seceng.informatik.tu-darmstadt.de
†Secure Mobile Networking Lab, TU Darmstadt, Germany, {mschmittner,lalmon,mhollick}@seemoo.tu-darmstadt.de
‡Dept. of Math. & Comp. Sci., Philipps-Universität Marburg, Germany, {lbaumgaertner,freisleb}@informatik.uni-marburg.de
*Co-first authors (these authors contributed equally to this work.)

***Abstract*—During disasters, existing telecommunication infrastructures are often congested or even destroyed. In these situations, mobile devices can be interconnected using wireless ad hoc and disruption-tolerant networking to establish a backup emergency communication system for civilians and emergency services. However, such communication systems entail serious security risks, since adversaries may attempt to steal confidential data, fake notifications of emergency services, or perform denial-of-service (DoS) attacks. In this paper, we present *SEDCOS*, a secure device-to-device communication system for disaster scenarios. *SEDCOS* mitigates flooding DoS attacks and offers role revocation for detected adversaries to withdraw their permissions. We demonstrate the effectiveness of *SEDCOS* by large-scale network simulations.**

***Index Terms*—Security, DoS, Disaster, D2D**

## I. Introduction

During floods, hurricanes, earthquakes, nuclear accidents, or terror attacks, fast disaster response can save human life, limit environmental damage, and reduce economic loss. Communication technologies are integral to disaster relief operations. However, panic reactions and physical damage often lead to inoperable local communication infrastructures [4].

As an attractive alternative to handheld radios, many researchers have proposed to leverage the ad hoc and *disruption-tolerant networking (DTN)* capabilities of mobile devices to create opportunistic communication networks [9], [12]. In DTNs, all devices store, carry, and forward data to form a dynamic, infrastructure-less, and self-organized network. Coverage is increased by adding more devices to the network. In particular, the approach can be applied to mobile commodity devices, such as smartphones, tablets, and laptops, which are ubiquitous and provide diverse ad hoc communication capabilities (e. g., Wi-Fi and Bluetooth). In this way, people can continue using their personal devices to request or offer aid, obtain information from emergency services, or contact relatives and friends.

However, such opportunistic networks are susceptible to a wide range of security attacks due to their wireless, cooperative, decentralized, and resource-constrained nature. For instance, during wars or terror attacks, adversaries may subvert the communication system to disrupt disaster relief operations by injecting false information or performing denial-of-service (DoS) attacks. Furthermore, panicked people may spam the network with messages, unintentionally jeopardize availability.

Thus, a practical emergency communication system must ensure confidentiality, authenticity, integrity, and availability, but these properties are difficult to achieve during adverse events. Existing proposals either lack disaster functionality or provide an insufficient level of security [3], [13], [14], [15]. High data availability and reliability are crucial for emergency notifications and distress signals. Prior work has improved reliability, but has not assessed secure prioritization mechanisms that work reliably under attack.

In this paper, we present *SEDCOS*, a secure device-to-device communication system for disaster scenarios. Our main contributions are: $(i)$ a secure communication substrate with message prioritization and a management scheme that delivers messages reliably and is resilient against flooding DoS attacks, and $(ii)$ large-scale network simulations showing *SEDCOS*'s effectiveness in maintaining high delivery rates under attack and revoking user certificates in the field.

## II. Related Work

Typical security targets in opportunistic networks are authentication and integrity of messages [17], secure routing [1], and confidential as well as anonymous end-to-end communication [5]. Identity-based Cryptography (IBC) is a frequently suggested solution, since traditional public key cryptography is often regarded as unsuitable for opportunistic networks due to the need of accessing public keys, certificates, and revocation information from central online servers [11]. To eliminate central authorities, fully decentralized trust-based concepts [2], [16], or approaches based on threshold-cryptography [10] have been proposed. However, existing works do not address the unique challenges of disaster relief communication, such as a high delivery rate (emergency messages), immobility of individual users (trapped victims), role-based authentication, or insider attackers. Denial-of-service attacks on unauthenticated DTNs have been evaluated, but contrary to previous findings [1], we show that authentication is essential for reliable operation. Other works have attempted to hinder flooding attacks by setting explicit rate limits and trying to detect misbehaving nodes using a complex distributed detection mechanism [8].

## III. System Model

**Store, Carry, and Forward.** Instead of relying on infrastructure, DTN-enabled devices exchange messages directly using WiFi and Bluetooth. DTNs exploit user mobility to increase coverage: devices act as "data mules" that store their messages as well as messages from other users, carry them, and finally forward them to the destination upon contact. This

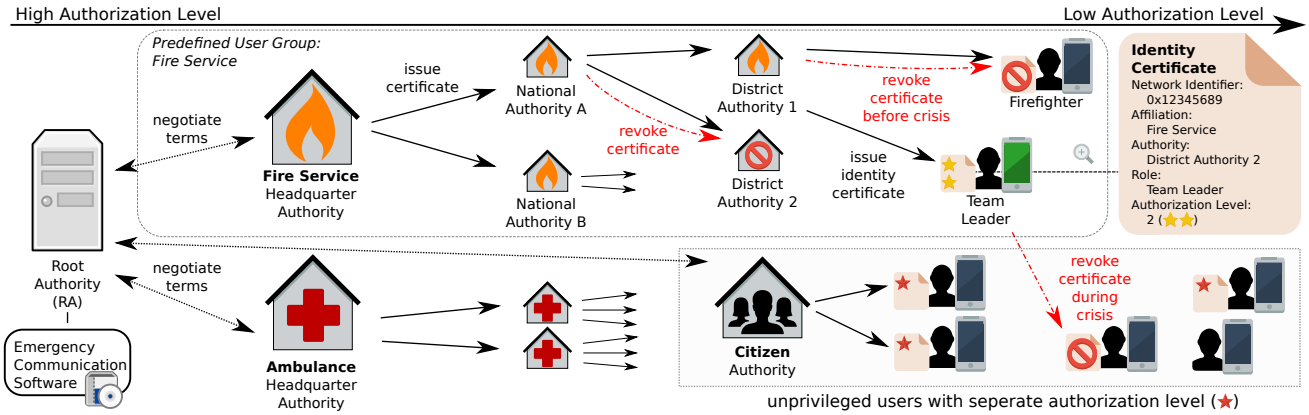



Figure 1: Illustration of our secure key management. The authorization level decreases from left to right, except for citizens.

way, messages propagate in an *epidemic* manner from device to device until they reach their destinations. DTN performance is typically worse than that of infrastructure networks but is preferable to no communication at all. Nevertheless, devices with Internet connectivity (cellular or WiFi access points) can opportunistically act as "wormholes" used for rapid message distribution to isolated parts of the network.

**Adversary Model.** We consider an adversary Adv who can mount network attacks and compromise network entities. Specifically, Adv can eavesdrop, manipulate, forge, or drop messages. Furthermore, Adv can assume a limited number of entities, either by compromising or stealing devices or by registering multiple times in our system. Unlike the classic Dolev–Yao adversary model, Adv controls only a part of the communication channel and a portion of all network entities. Moreover, Adv cannot break cryptographic primitives or tamper with the *root authority* (see Section IV).

## IV. Secure Key Management

Establishing trust is important to satisfy our security requirements. For this purpose, we employ a centralized trust model using a Public Key Infrastructure (PKI), which is shown in Fig. 1. The PKI consists of multiple hierarchically organized certificate authorities (CAs), whose root is a dedicated authority named *root authority (RA)*. The RA serves as a trust anchor, maintains the emergency communication software, and distributes the software if infrastructure access is still available. In an initialization phase, before the actual crisis, the RA establishes relationships to organizations or governments that want to participate as *authorities* in the emergency communication system. All authorities initially undergo a rigorous audit by the RA, since their authenticity and trustworthiness are crucial to the overall security. As part of the audit process, RA and authority agree on *user roles* as well as preconfigured *user groups* that the authority introduces to the network. For instance, in Fig. 1, the fire service organization added the user roles *team leader* and *firefighter*, and arranged a preconfigured user group *fire service*, so users can particularly address all firefighters when sending a message. Authorities manage their own PKI and, hence, maintain one or multiple, potentially hierarchically organized, CAs. The CAs' public keys are embedded in the emergency communication software. After this step, authorities can issue *identity certificates*. Furthermore, the overall PKI contains at least one authority that issues identity certificates to *unprivileged* users, i.e., citizens. In Fig. 1, the fire service maintains several hierarchically organized authorities. On the lowest hierarchical CA level, CAs issue identity certificates to staff members.

Identity certificates bind the public signing keys of users, which function as their unique network identifiers (see Section V-A), to user properties. A vital user property is the *user role*, since it is important to assess the content of messages. For instance, citizens consider medical information more reliable if they originate from physicians rather than firefighters. Another essential property is the *authorization level* that indicates the permission level and trustworthiness of a user. Fig. 1 shows the identity certificate of a firefighter team leader, and depicts the authorization level of entities by their x-coordinate as well as stars in the certificate. In order to obtain an identity certificate, users must register with the CA and provide a proof of identity, e.g., using their identification card, phone number, or address. The identity proof is vital to hamper multi-registrations, where a single user obtains multiple identity certificates.

Since an adversary may obtain identity certificates, compromise user devices, or even infiltrate authorities, it is important that certificates can be revoked. *SEDCOS* implements certificate revocations via certificate revocation lists (CRLs) that are broadcasted with high priority in the network. We distinguish between two different entities: authorities and users. An authority $\mathcal{A}$ can revoke an entity $\mathcal{E}$ if $\mathcal{A}$ has a higher authorization level than $\mathcal{E}$, and there is a certificate chain (i.e., a chain of trust) between $\mathcal{A}$ and $\mathcal{E}$. Upon the revocation of an authority, all certificates that the authority issued in the past and will issue in the future are regarded as invalid, withdrawing its power. In case a user identity certificate is revoked, the respective user becomes an uncertified and unprivileged user, hence, loses its user role, authorization level, and any message transmission privileges (see Section V-B).

## V. Resilient Communication

In this section, we first give an overview of our communication protocol and then explain the design of our DoS-resistant buffer management.

### A. Protocol Overview

**Message Format and Types.** All *SEDCOS* messages have the same format and include the following fields: message

*type*, sender and receiver *addresses*, *creation time* and *lifetime* (together yielding the time-to-live (TTL)), sender *signature*, and the optionally encrypted *payload*. We note that all header fields are *immutable*, that is, they are not changed in transit, thus, allowing the signature to protect the entire message. The message type can be: ($i$) *Certificate Revocation Lists*; ($ii$) *Network control* with subtypes for acknowledgments and the device-to-device message exchange handshake; ($iii$) *Content* sent by users. Acknowledgments are sent by the destination upon reception of a message.

**Message Authenticity and Confidentiality.** Each user possesses a unique Elliptic Curve Digital Signature (ECDSA) *signature key pair*. The public part of the key serves as a unique addressable *network identifier*. Each outgoing message is signed using this key and can optionally be augmented with the identity certificate. Devices verify messages at each hop by checking the message signature and, if available, the sender's identity certificate; and discard them if any check fails. Hence, corrupted messages do not propagate in the network. To achieve data confidentiality, each user generates its own Elliptic Curve Integrated Encryption Scheme (ECIES) *encryption key pair* during initialization. Consequently, sending or receiving confidential messages requires the message payload to be encrypted with the public or decrypted with the private ECIES key of the receiving user.

**Message Storage.** Each device reserves persistent memory for storing its own as well as others' messages. We refer to this memory space as the *buffer*. Its capacity $C$ depends on the device capabilities and can be adjusted by the user. Efficiently managing the buffer is crucial for delivery reliability, as shown in Section VI.

**Message Exchange.** When two devices discover each other via Bluetooth or Wi-Fi beacon frames, they connect to exchange messages. We use *epidemic* dissemination, i.e., nodes try to exchange all carried messages. This introduces redundancy in the network, which helps when single nodes "disappear" (low battery or mobility). However, due to limited buffer capacity and possibly short contact times (i. e., two cars passing each other), not all messages might be exchanged. Thus, we exchange messages in the following order: ($i$) messages destined for $\mathcal{B}$, ($ii$) messages from *privileged* users (authorities), ($iii$) all other messages.

### B. Source-based Elastic Buckets

Proper buffer management is essential to prevent resource starvation attacks such as flooding. Malicious nodes can easily exploit trivial implementations such as FIFO queues containing all messages to replace valid messages with bogus ones [7]. To counter such attacks, we employ a novel buffer management strategy called *Source-based Elastic Buckets (SEB)* that, by design, prevents valid messages from being purged during flooding attacks. The basic idea is that all messages from a source $s$ are placed in an isolated bucket $B$ such that messages from different sources cannot influence one another. SEB is fair in the sense that each bucket has a guaranteed *capacity* of $C_B = \lfloor C/n \rfloor$ where $n$ is the number of currently allocated buckets (= number of source nodes we currently carry messages from). The *occupancy* of a single source bucket $O_B$ is subject to $O_B \in [0, C]$ and $\sum_s O_B \leq C$. If $s$ does not exhaust its guaranteed capacity ($O_B < C_B$) because it has not sent "enough" messages, free capacity ($C_B - O_B$) is shared by other buckets requiring it. However, when $s$ sends a message at a later point, overdrawn buckets ($O_B > C_B$) are emptied first. These *elastic* quotas allow full exploitation of local buffer capacities while maintaining strict message separation of different source nodes. Algorithm 1 shows SEB's message insertion procedure: the underlying idea is that SEB inserts new messages in the appropriate (source) bucket and then drops messages from the highest occupant bucket until the total occupancy meets $C$. Note that a node will always try to make space for its messages by dropping its messages last. This is to ensure that there is at least one copy of every message in the network. However, if a device injects too many new messages (exceeding $C$), its buffer overflows, and SEB eventually has to drop own messages. Within each bucket, SEB prioritizes: ($i$) security control messages (revocation certificates), ($ii$) network control messages (acknowledgments), and ($iii$) messages with the longest remaining TTL.

**Algorithm 1** Source-based Elastic Bucket Insertion

**Input:** $msg$, $buckets$, $C$
1: $s \leftarrow$ source of $msg$
2: **if not** $buckets$ contains bucket for $s$ **then**
3: insert new empty bucket for $msg$ in $buckets$;
4: **end if**
5: $B_s \leftarrow$ bucket from $buckets$ for $s$;
6: insert $msg$ into $B_s$;
7: **while** occupancy of $buckets$ exceeds $C$ **do**
8: $\hat{B} \leftarrow$ bucket from $buckets$ with the highest occupancy;
9: remove message with the lowest rank from $\hat{B}$;
10: **if** $\hat{B}$ is empty **then**
11: remove $\hat{B}$ from $buckets$;
12: **end if**
13: **end while**

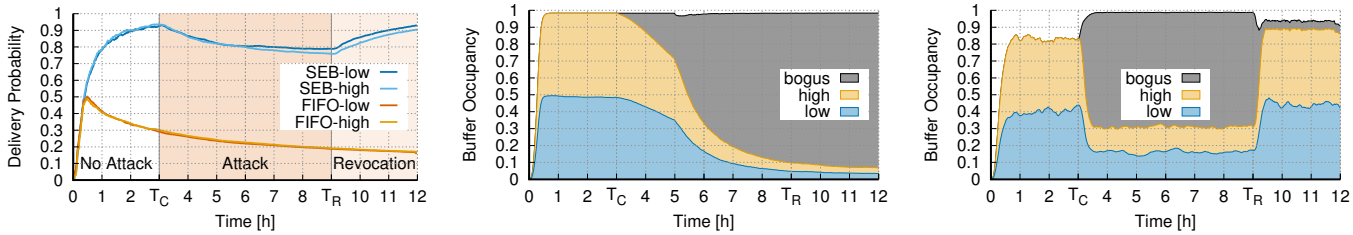
SEB's robustness relies on the fact that messages are source-authenticated and on the relatively high costs of acquiring new identities in our system. Without the latter costs, an attacker could assume multiple identities, flood the network with messages and, thus, hijack a disproportional amount of buffer capacity.

## VI. Experimental Evaluation

In this section, we evaluate the impact of flooding attacks by several privileged devices (due to theft or compromise), and their eventual revocation from the system.

**Scenario.** We consider three different user classes with a total of 1000 nodes: 850 *citizens*, 100 *authorities*, and 50 *attackers*. Within each group, there are 5 % cars (10–50 km/h), all others move at walking speed (1.8–4.5 km/h). Citizens can transmit *low*-priority messages, while authorities sent with *high* priority (interval: 15–25 s). We compare *SEB* to a classic *FIFO* queue, both using epidemic routing. The buffer capacity $C$ is 5 MB. We use the *ONE* simulator v1.6.0 [6] as well as the default Helsinki map for our experiments and average the results over ten differently-seeded runs. We assume Bluetooth communication with 2 Mbit/s and a range of 10 m.

**Flooding Attack and Revocation.** We evaluate the impact




(a) Delivery rate of valid messages over time. (b) Buffer occupancy with *FIFO* (c) Buffer occupancy with *SEB*

Figure 2: **Flooding attack and revocation.** The attack starts at $T_C = 3\,\text{h}$ and the revocation certificate is issued at $T_R = 9\,\text{h}$.

of an attacker being able to compromise privileged devices. In this case, the attacker is able to inject *bogus* high-priority messages into the network, thus, increasing their chances to remain in the nodes' buffers for a long time. In this experiment, two events occur: at $T_C = 3\,\text{h}$, the attackers start the flooding attack using compromised devices; and after a reaction time of $6\,\text{h}$ at $T_R = 9\,\text{h}$, an authority issues and injects the revocation certificate into the network. When a node receives the revocation certificate, it drops all messages it carries from the revoked nodes and blacklists future messages by those nodes. Attackers ignore revocation certificates.

*Before the attack.* Figure 2a shows the delivered benign messages (*low* and *high*) over time. In the beginning, *SEB* quickly starts to successfully deliver most messages. *FIFO* follows the same start-up behavior but is not able to keep up from the 30 minutes mark. After 30 minutes, buffers are filled up (Fig. 2b) and the lack of proper buffer management leads to poor delivery performance.

*During the attack.* At the start of the attack, *SEB*'s delivery probability remains almost unaffected by the flooding attack even though buffers quickly fill up to 70 % with *bogus* messages (Fig. 2c). The decrease in delivery probability is only about 10 % (Fig. 2a) demonstrating the effectiveness of the source-based elastic buckets: they assure that *bogus* messages cannot overtake the entire buffer capacity. *FIFO* reacts less visibly to the attack since the delivery probability is already low at $T_C$ (Fig. 2a). Yet, the impact is apparent in Fig. 2b where bogus messages steadily take up more buffer capacity, leading to continuously decreasing delivery probability.

*Aftermath. FIFO* does not recover from the attack after $T_R$, since only a few nodes receive the revocation certificate due to the lack of a prioritization mechanism. With *SEB*, the revocation certificate propagates quickly throughout the network: half of the nodes are informed within 12 minutes while full network penetration is reached in less than one hour. The drop of bogus message buffer occupancy shortly after $T_R$ indicates the revocation certificate's effect (Fig. 2c). As the certificate propagates in the network, buffer occupancy restores to a state similar to $t < T_C$. Nevertheless, a small fraction of bogus messages remains in the network since attackers ignore the revocation certificate and keep their messages in their buffers.

## VII. Conclusion

A secure and reliable communication system is essential for effective disaster response. We have presented *SEDCOS*, a system that enables secure and reliable disruption-tolerant emergency communication on commodity mobile devices. *SEDCOS* is the first secure emergency communication system that enables the exclusion of adversaries while providing authentic and confidential group communication. Under DoS attacks, *SEDCOS* increases the message delivery rate by a factor of 6 compared to a contemporary DTN protocol. Finally, *SEDCOS* provides a timely revocation (less than one hour) for withdrawing any power of an insider adversary.

## Acknowledgment

This work is funded by the LOEWE initiative (Hessen, Germany) within the NICER project.

## References

[1] J. Burgess *et al.*, "Surviving attacks on disruption-tolerant networks without authentication," in *ACM MobiHoc*, 2007.
[2] R. Chen *et al.*, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1200–1210, 2014.
[3] T. Hossmann *et al.*, "Twitter in disaster mode: security architecture," in *ACM Special Workshop on Internet and Disasters*, 2011.
[4] IRIN News, "Life-saving radio begins broadcasting in typhoon-hit Tacloban," nov 2013. Online: http://www.irinnews.org/report/99132/life-saving-radio-begins-broadcasting-typhoon-hit-tacloban
[5] A. Kate *et al.*, "Anonymity and security in delay tolerant networks," in *IEEE SecureComm*, 2007.
[6] A. Keränen *et al.*, "The ONE simulator for DTN protocol evaluation," in *ICST SIMUTools*, 2009.
[7] F. C. Lee *et al.*, "A queuing mechanism to alleviate flooding attacks in probabilistic delay tolerant networks," in *IEEE AICT*, 2010.
[8] Q. Li *et al.*, "To lie or to comply: defending against flood attacks in disruption tolerant networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 3, pp. 168–182, 2013.
[9] Z. Lu *et al.*, "Networking smartphones for disaster recovery," in *IEEE PerCom*, 2016.
[10] J. Luo *et al.*, "Dictate: Distributed certification authority with probabilistic freshness for ad hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 4, pp. 311–323, 2005.
[11] D. Ma and G. Tsudik, "Security and privacy in emerging wireless networks [invited paper]," *IEEE Wireless Communications*, 2010.
[12] A. Martín-Campillo *et al.*, "Evaluating opportunistic networks in disaster scenarios," *Journal of Network and Computer Applications*, 2013.
[13] E. A. Panaousis *et al.*, "Secure decentralised ubiquitous networking for emergency communications," in *IEEE TEMU*, 2012.
[14] M. Puzar *et al.*, "Security and privacy issues in middleware for emergency and rescue applications," in *IEEE PervasiveHealth*, 2008.
[15] S. G. Weber *et al.*, "Mundomessage: Enabling trustworthy ubiquitous emergency communication," in *ACM IMCOM*, 2011.
[16] H. Zhu *et al.*, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.
[17] H. Zhu *et al.*, "An opportunistic batch bundle authentication scheme for energy constrained DTNs," in *IEEE INFOCOM*, 2010.