Suicide for the Common Good: a New Strategy for Credential Revocation in Self-Organizing Systems

Jolyon Clulow and Tyler Moore
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, United Kingdom
firstname.lastname@cl.cam.ac.uk

ABSTRACT

We consider the problem of credential revocation in self-organizing systems. In the absence of a common trusted authority, reaching a decision is slow, expensive and prone to manipulation. We propose a radical, new strategy—*suicide for the common good*—which drastically simplifies the decision-making process and revocation orders. Our mechanism is fully decentralized, incurs low communication and storage overhead, enables fast removal of misbehaving nodes, and is ideally suited to highly mobile networks.

1. INTRODUCTION

In medieval Japan, the warrior class of the samurai was structured according to a rigid code of honor. Loyalty to the clan was absolute: the daimyo (feudal lord) had power of life and death over his subjects. His samurai would readily commit seppuku (ritual suicide) to demonstrate their unquestioning loyalty, accept responsibility for a mistake, or clear the name of their clan. Our insight is that this principle of valuing the clan more than one's own existence, to the point of being ready to commit suicide in the interest of the common good, may be profitably applied to self-organizing ad-hoc networks.

In this note, we consider the task of revoking credentials within a decentralized network. This can happen whenever a key has been compromised or a node has been identified as misbehaving.

In a centralized network, revocation is a straightforward and well-studied problem. For systems using public keys, a trusted certification authority (CA) periodically issues a certificate revocation list (CRL) of revoked nodes or secrets [11]. It is up to the CA to decide when certificates are to be revoked. Because all nodes trust the CA, there is little worry of false accusations.

In a self-organized system, by contrast, no clear, universally accepted authority exists, so the revocation process becomes much more difficult (and costly) to implement in practice. This is because it is unclear how to *decide* whether a device should be revoked. A malicious participant can falsely accuse another node of misbehavior. Even if a decision can be reached, revoking credentials remains challenging since nodes do not necessarily *trust* claimed results from the vote.

We propose a radical strategy, *suicide for the common good*, which drastically simplifies the decision-making process and revocation orders. It allows a single node to revoke a malicious node from the network at the cost of its own membership. This strategy mimics a number of mechanisms prevalent in nature, such as the sacrificial action of bees defending the hive by stinging an intruder and dying as a result. Our immediate goal is not to describe a fixed protocol but to present and analyze this new strategy, highlighting the necessary conditions to make it viable. Our revocation proposal exhibits a number of desirable properties: it is fully decentralized, incurs low communication and storage overhead, enables fast removal of misbehaving nodes, and is ideally suited to highly mobile networks.

2. EXISTING DECISION MECHANISMS

Surprisingly few protocols deal with the issue of revocation in decentralized networks. Typically, revocation strategies are presented in combination with proposals for detecting anomalous behavior or for key distribution. However, attack detection is an independent prerequisite for credential revocation: in principle, any revocation technique can be used in combination with the chosen detection mechanism.

Existing proposals for collective decision-making in self-organizing networks are predominately voting-based. The underlying premise has been that since no node has system-wide authority, everyone should have an equal say in deciding when credentials should be revoked. Such voting has manifested itself in several ways, from direct majority-voting schemes to elaborate reputation systems that collect historical ratings of interactions.

As noted in [20], reputation systems are effectively voting schemes, since the aggregated opinions of many nodes produce rankings to punish misbehaving nodes through isolation. The idea behind reputation systems is that because nodes may interact rarely or even just once, a collective, shared history can more efficiently identify malicious nodes. Proposed decentralized reputation systems have aimed to deter free-riding in peer-to-peer systems [13, 8, 21] and routing misbehavior in wireless ad hoc networks [2, 16, 9].

A threshold protocol takes a binary decision based on whether the required number of votes has been cast. Sometimes the task of tallying votes remains centralized, for example, done by the base station [15]. The truly distributed approach has been to use threshold cryptography to split the task of signing certificates [22, 14, 6]. In this way, nodes can issue certificate revocation lists signed by a threshold of users. In [5], Chan, Perrig and Song present a threshold voting revocation scheme specifically designed for pairwise symmetric key pre-distribution schemes in low-energy sensor

networks. Their scheme is extended and generalized in [4]. Notably, these schemes are limited to immobile networks as only the immediate neighbors of a node at network initialization are given the ability to vote on that node's behavior.

Yet collective decision-making is slow, expensive and prone to manipulation. Disadvantages of voting-based schemes include:

- **Susceptibility to false accusations** Attacker-controlled nodes can transmit votes against legitimate nodes without consequence. This undermines the credibility of votes from honest nodes.
- **Susceptibility to collusive attackers** Attacker-controlled nodes can concentrate themselves in a chosen location to create a local majority.
- **Susceptibility to Sybil and replication attacks** Attacker-controlled nodes can rig votes with spurious identities [7] or by re-using an identity in multiple locations.
- **Susceptibility to selective misbehavior** As noted in [5], threshold voting schemes are vulnerable to an attacker who reveals detectable misbehavior to just fewer nodes than the number needed to initiate revocation.
- **Slow attack response** Since no single node's claim can be trusted, significant time may pass, and attacks may be launched, before a voting scheme triggers a revocation order.
- **High storage and communications overhead** Threshold voting and reputation systems impose significant storage and computational requirements.

3. SUICIDE FOR THE COMMON GOOD

Reaching decisions can be made much simpler if we allow a single node to decide. If a node believes another has misbehaved, then it can carry out punishment. The trouble with this approach is that a malicious node can falsely accuse legitimate ones; the solution is to make the act of punishment costly. We propose a simple, albeit radical strategy: *suicide for the common good*.

Upon detecting a node M engaging in some illegal activity, A broadcasts a signed *suicide note* which includes the identities of both A and M. The other nodes in the network then verify the signature and, if correct, revoke both A and M. This can be achieved by adding both identities to a blacklist and deleting all keys shared with either node. This strategy is premised on the observation that if a node determines another node has cheated, there is no more convincing way to let its neighbors know of its sincerity than to transmit a signed self-revocation certificate.

The suicide note $A, M, \mathrm{sig}_K(A, M)$ described in the protocol fragment given in Figure 1 can be implemented using either public key or symmetric key cryptography. For public key infrastructures, the signing key K is node A's private signing key K_A^{-1} and other nodes verify the signature using the public verifying key K_A. We assume the existence of identity-based cryptography or public key certificates so that nodes can easily determine public keys. The same signed suicide note can therefore be broadcast throughout the network for all to verify. We assume, as voting mechanisms do, that the network remains connected as a single component.

$$
\begin{array}{rl}
A: & \text{detects } M \text{ misbehaving} \\
A \longrightarrow *: & A, M, \mathrm{sig}_K(A, M) \\
*: & \text{verifies signature and adds } A, M \text{ to blacklist}
\end{array}
$$

Figure 1: Suicide for the common good protocol fragment.

The situation changes slightly for a symmetric key-based solution. Unfortunately any key shared between more than two nodes, such as a group key, cannot be be used as this would allow fraudulent suicide notes to be created. A practical solution is to use TESLA authentication keys [19] where the suicide note is signed with a key that is released after all nodes have received a copy of the note. During the predistribution phase, the network owner generates a unique revocation key for each node. The owner calculates a hash tree with the hash of the key and a node identifier ($h(i, k_i)$) as leaves. Every nodes stores a copy of the resulting root authentication value. Each node also stores its unique revocation key and associated $\log n$ intermediate authentication values. To revoke a target node, a node signs the suicide note which is then broadcast throughout the network. Then it reveals its revocation key and intermediate authentication values.

To be effective, the suicide strategy must ensure that multiple nodes do not issue certificates for a single misbehaving node. Multiple claims can arise under two circumstances: from nodes in one location area or from nodes in different areas.

In the former case, two honest nodes observe a misbehaving node and simultaneously issue a suicide note. The likelihood of happening can be reduced by incorporating a random delay or back-off before issuing suicide notes. However, this solution is not absolute and could still cause multiple notes to be issued. It can be extended to a two-pass mechanism where a timestamped *offer* is first broadcast. After a suitable delay, the earliest stamped offer is accepted, and that node issues the final suicide note.

The latter case arises whenever a node presents itself in several locations, either re-using identities (node replication) or presenting different ones (Sybil). We assume that orthogonal mechanisms exist for detecting and preventing Sybil attacks (e.g., [17]) and node replication attacks (e.g., [18]).

4. DISCUSSION AND ANALYSIS

4.1 Properties

Suicide for the common good exhibits several appealing properties compared to other decision-making schemes:

1. **Low communications overhead** No need to send messages back and forth between voting members.
2. **Fully decentralized** No need to consult a base station.
3. **Very fast removal** No delays while waiting on votes or thresholds to be met.
4. **Unconstrained node mobility** Nodes are not restricted to a location or set of voting neighbors.

5. **Undermines false claims as attack strategy** False claims remove only one innocent node.

6. **Single node detection** Only one honest node has to detect misbehavior to initiate revocation.

4.2 Performance analysis

Suicide for the common good is an extremely efficient decision mechanism. It avoids the overhead associated with normal voting or consensus schemes by allowing a single node to decide. For comparison, Chan *et al.*'s scheme [5] requires $O(d \cdot \log(d))$ storage and $O(d)$ communications per node (where d is the number of voting members) to reach a decision. In contrast, a single signature generation operation is required to produce the suicide certificate. The note must be broadcast to all nodes in the network. If nodes only shares keys with their immediate neighbors, then this cost can be significantly reduced. Upon receiving the certificate, each node must perform one signature verification.

We note that the signature generation operation is the last computation performed by a node committing suicide. Public key cryptography in constrained environments such as sensor networks is often criticized because of the low speed and high energy consumption associated with private key operations (e.g., signing messages). But gratuitous energy consumption is of little consequence here since the node is removing itself from the network; other nodes only perform less costly signature verification. Likewise, the delay associated with signing the suicide note is largely inconsequential since it does not slow down normal communications. At worst, the slight lag in revoking the node allows the attacker a short stay of execution. Moreover, this delay is insignificant compared to that associated with consensus or voting mechanisms. Thus suicide is ideally suited to a hybrid key infrastructure where symmetric key operations dominate and public key operations are limited to specific instances where they provide useful security properties.

4.3 Conditions for suicide

We note that suicide for the common good is an effective decision-making strategy only when certain conditions are met:

1. Attacker benefit from removing one innocent node must be less than the benefit of having a malicious node placed inside the network.

2. Honest nodes share common interest.

3. An absence of unforgeable, independently verifiable and conclusive proof.

4. Low likelihood of two good nodes accusing each other.

5. Difficult to prevent malicious nodes from issuing false claims.

Condition 1 can be met whenever the number of good nodes dominates the number of bad nodes present in the system. In addition, the value of nodes must be consistent: a smart dust mote must not be able to revoke a base station, for instance. When the condition is met, we can afford to sacrifice a good node for the benefit of removing a bad node. This is a strategy employed in nature (e.g., white blood cells in macrophage). The suicide scheme could be extended to more general ratios reflecting the relative value of nodes(e.g., requiring two nodes to be sacrificed in order to remove one bad node).

One threat is that adversaries may use a node's removal to disrupt a valid route or create a numerical advantage in an area. In principle, the adversary should not be able to influence network topology. To mitigate this threat, we propose that reinforcements be sent to repopulate an attacked area. This is a natural response since a series of suicides in a region indicates likely enemy action. Thus we can probabilistically move nodes closer to the area where the suicide note has been issued.

Condition 2 requires honest nodes to value the social welfare of the network over individual utility. This condition is reasonable whenever the nodes are deployed by a single entity (e.g., a sensor network deployed on a battlefield) as opposed to when nodes are individually controlled (e.g., a peer-to-peer file-sharing system).

While conditions 1 & 2 are system dependent, conditions 3–5 depend on the corresponding detection mechanism. Hence the detection mechanism impacts the choice of the most appropriate revocation strategy. Occasionally, evidence of misbehavior is non-repudiable to any third party, which normally requires digital signatures and asymmetric cryptography. Detection mechanisms producing universally-verifiable evidence include geographic packet leashes [12] for detecting wormholes and node replication detection in sensor networks [18]. For these schemes, suicide for the common good is inappropriate, since the malicious node's guilt is incontrovertible and verifiable to all without the need for consensus (contradicting condition 3).

However, generating universally non-repudiable evidence can be costly, implicitly requiring widespread use of public key cryptography and broadcasting many signed messages. Furthermore, situations where a malicious node is forced into self-incrimination are limited. This excludes, for instance, detecting a malicious node that chooses to not do something such as dropping a message.

More commonly, detection mechanisms create evidence that is non-repudiable only to a single party. This can happen for evidence signed using a pairwise unique symmetric key, for instance. A message authentication code (MAC) guarantees origin authenticity to the nodes who hold the signing key. If pairwise symmetric keys are used, then the two nodes sharing the key know the message is authentic; however, no other nodes are so assured. Detection mechanisms of this type include temporal packet leashes [12], Sybil attack [7] detection by querying for possessed keys [17] and distance-bounding protocols [1, 10, 3].

Such detection mechanisms are amenable to suicide. Malicious nodes cannot trick honest nodes into falsely accusing each other without knowing the relevant key (satisfying condition 4). Yet dishonest nodes can easily levy false accusations because evidence is not universally verifiable (satisfying condition 5).

5. FUTURE WORK

We have introduced a new strategy for credential revocation in self-organizing systems; in future work we plan to specify a complete, formal protocol. In particular, we would would like to pursue a more rigorous security analysis, considering partitioning, replication and Sybil attacks. We can also conceive of a scenario where a central authority is used for distributing suicide notes while keeping the detection

fully distributed. Further extensions include incorporating inconclusive evidence and uncertainty from detection mechanisms and generalizing the scheme to where x nodes offer themselves to revoke y nodes.

6. CONCLUSION

We have presented *suicide for the common good*, an effective and efficient credential revocation strategy for self-organizing systems. Suicide for the common good compares favorably to existing voting-based revocation mechanisms in terms of speed, communications overhead and storage requirements. Furthermore, to the best of our knowledge, it is the first fully decentralized revocation strategy that works even when nodes are highly mobile. We hope that future work will identify more applications and present formal specifications of secure protocols to realize these ideas.

7. ACKNOWLEDGMENTS

The authors would like to thank Ross Anderson, Richard Clayton, Markus Kuhn and Frank Stajano for their helpful comments; the authors additionally thank Frank Stajano for discussing the history of seppuku.

8. REFERENCES

[1] S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *EUROCRYPT*, pages 344–359, 1993.

[2] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol. In *MobiHoc*, pages 226–236. ACM, 2002.

[3] S. Capkun, L. Buttyán, and J.-P. Hubaux. Sector: secure tracking of node encounters in multi-hop wireless networks. In S. Setia and V. Swarup, editors, *SASN*, pages 21–32. ACM, 2003.

[4] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Trans. Dependable Secur. Comput.*, 2(3):233–247, 2005.

[5] H. Chan, A. Perrig, and D. X. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages 197–. IEEE Computer Society, 2003.

[6] C. Crépeau and C. R. Davis. A certificate revocation scheme for wireless ad hoc networks. In *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 54–61, New York, NY, USA, 2003. ACM Press.

[7] J. R. Douceur. The Sybil attack. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, *IPTPS*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

[8] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111. ACM Press, 2004.

[9] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In S. Setia and V. Swarup, editors, *SASN*, pages 66–77. ACM, 2004.

[10] G. P. Hancke and M. G. Kuhn. An RFID distance bounding protocol. In *IEEE SecureComm 2005, Athens, Greece, 5–9 September 2005*, pages 67–73. IEEE Computer Society, 2005.

[11] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280 (Proposed Standard), Apr. 2002. Updated by RFC 4325.

[12] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *INFOCOM*, 2003.

[13] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *WWW '03: Proceedings of the twelfth international conference on World Wide Web*, pages 640–651. ACM Press, 2003.

[14] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *ICNP*, pages 251–260. IEEE Computer Society, 2001.

[15] D. Liu, P. Ning, and W. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *ICDCS*, pages 609–619. IEEE Computer Society, 2005.

[16] P. Michiardi and R. Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In B. Jerman-Blazic and T. Klobucar, editors, *Communications and Multimedia Security*, volume 228 of *IFIP Conference Proceedings*, pages 107–121. Kluwer, 2002.

[17] J. Newsome, E. Shi, D. X. Song, and A. Perrig. The Sybil attack in sensor networks: analysis & defenses. In K. Ramchandran, J. Sztipanovits, J. C. Hou, and T. N. Pappas, editors, *IPSN*, pages 259–268. ACM, 2004.

[18] B. Parno, A. Perrig, and V. D. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pages 49–63. IEEE Computer Society, 2005.

[19] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. SPINS: security protocols for sensor networks. *Wirel. Netw.*, 8(5):521–534, 2002.

[20] A. Serjantov and R. Anderson. On dealing with adversaries fairly. In *Proceedings of the Third Annual Workshop on Economics and Information Security*, May 2004.

[21] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *ACM Conference on Electronic Commerce*, pages 228–229. ACM, 2003.

[22] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.