

Context-Enhanced Authentication for Infrastructureless Network Environments

Ryan Wishart^{1,*}, Jadwiga Indulska^{1,2}, Marius Portmann^{1,2}, and Peter Sutton¹

¹ School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, Australia

{wishart, jaga, marius, p.sutton}@itee.uq.edu.au

² National ICT Australia **

Abstract. Infrastructureless networks are becoming more popular with the increased prevalence of wireless networking technology. A significant challenge faced by these infrastructureless networks is that of providing security. In this paper we examine the issue of authentication, a fundamental component of most security approaches, and show how it can be performed despite an absence of trusted infrastructure and limited or no existing trust relationship between network nodes. Our approach enables nodes to authenticate using a combination of contextual information, harvested from the environment, and traditional authentication factors (such as public key cryptography). Underlying our solution is a generic threshold signature scheme that enables distributed generation of digital certificates.

1 Introduction

Infrastructureless network environments, including both Mobile Ad hoc Networks (MANET) and many pervasive computing environments, have enjoyed increased attention of late. While the lack of fixed infrastructure in these networks makes them quick to deploy, it also presents a problem from a security perspective.

Within this paper we focus on one particular aspect of security, authentication, and provide a solution that overcomes many of the problems of the infrastructureless environment that have hampered previous approaches. These previous authentication approaches have typically assumed that the network is a region under the control of a centralised authority. This centralised authority shares secret knowledge with all of the nodes that are permitted entry to the network, and can use that knowledge to authenticate the nodes. This secret knowledge may

* The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources).

** National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

take the form of a secret password, as in Kerberos [1], knowledge of specialised hardware possessed by the node (such as a smart card) or defining biometric characteristics (when authenticating people).

Problems arise when these traditional authentication approaches are applied to infrastructureless networks as (1) there is a lack of trusted infrastructure, (2) the network size is often highly dynamic, and (3) the network may consist of nodes that have never encountered one another before and thus have no knowledge of one another to base the authentication procedure on.

In this paper we present a multi-factor, context-enhanced authentication mechanism for infrastructureless network environments. The mechanism consists of a distributed collection of mutually authenticated nodes that can check particular authentication factors and are trusted to do so. These factors can be either traditional factors such as digital certificates, or contextual factors (such as location). By supporting contextual factors, our mechanism can perform authentication of new nodes in situations where the new node and the authentication mechanisms nodes have no shared knowledge. To generate certificates in the absence of a central Certification Authority, our approach uses a threshold signature generation scheme.

The remainder of this paper is organised as follows. Section 2 provides background information on the threshold signature generation technique underlying our approach. Section 3 surveys related work in the field of authentication with particular attention given to mechanisms intended for Mobile Ad hoc Networking (MANET) and pervasive computing environments. The requirements for an underlying generic threshold signature scheme (TSS) are outlined in Section 4. In Section 5 the types of context information that can be included in the authentication process are discussed. The operation of our authentication mechanism is discussed in Section 6. Extensions to the authentication mechanism to permit grouping of nodes based on the factors they used during authentication are then covered in Section 7. An evaluation of our approach is discussed in Section 8 before concluding remarks are given in Section 9.

2 Threshold Signature Generation

Threshold signature generation techniques permit a group of entities (which we shall refer to as secret share holders) to distributedly generate a digital signature on a certificate without the need for a central Certification Authority. To achieve this, the private key to be used for the signature generation is split into a number of “secret shares” by a trusted dealer. One secret share is then given to each secret share holder [2].

To generate a digital certificate it is necessary for unsigned copies of the certificate to be distributed to a threshold number of the secret share holders. Secret share holders that receive a certificate then apply their secret share to their copy of the certificate. This generates a partial signature on the copy of the certificate. These partially signed copies of the certificate can then be combined together to generate a valid digital certificate. The task of combining the partially

signed certificates into a valid digital certificate can be performed either by a dedicated “combiner” node, or by the node requesting the certificate (as in [3]).

Several developments can be applied to threshold signature schemes to increase their usefulness. These developments include eliminating the need for a trusted dealer through Joint Secret Sharing [4], and a means to check if partial signatures are valid (referred to as Verifiable Secret Sharing [5]). A third development, Proactive Secret Sharing [6], enables distributed refreshing of secret shares. Refreshed secret shares are incompatible with previously used secret shares.

3 Related Work

Authentication mechanisms for networks with fixed infrastructure are not easily migrated to infrastructureless network environments as they often depend on the availability of trusted infrastructure, as is the case with the widely used Kerberos [1] protocol.

Pirzada and McDonald [7] developed a distributed implementation of Kerberos for MANET environments that replicates the Kerberos Authentication Server across a group of highly trusted nodes. In an infrastructureless environment, where all nodes are potentially strangers, locating highly trusted nodes is likely to be problematic.

A different approach is taken by Zhou and Haas [8] who suggest using a distributed Certification Authority (based on Shamir’s secret sharing [2]). However, they consider only traditional authentication mechanisms, and their approach cannot cope with nodes previously unknown to the system. Distributed signature generation schemes are used by Saxena et al. [9] and Luo et al. [10] for admission control to MANET networks. Neither of these two papers discuss their supporting authentication in detail. However, they do show that a signature generation scheme can be used in the MANET environment for security purposes.

Glynos et al. [11] present a mechanism for authentication in MANET environments that combines traditional authentication with a limited set of context information used to identify MANET nodes. The context information is compared against profiled information to establish MANET node identity.

A similar approach is taken by Covington [12], although Covington does not support traditional authentication mechanisms. A major failing with both Glynos et al. and Covington’s solutions is that they cannot easily be used in an infrastructureless network as they require the creation of context profiles for each node that may join the network.

4 Requirements for the Underlying Threshold Signature Scheme

Our approach to context-enhanced authentication in infrastructureless environments has been designed to operate on top of a generic threshold signature scheme (TSS). In this section we discuss the requirements our design places on this underlying TSS. The specific requirements are that the TSS should:

1. be cryptographically secure
2. handle fluctuating network size
3. support Joint Secret Sharing so that a trusted dealer is not required
4. support Verifiable Secret Sharing techniques to identify invalid partial signatures
5. support Pro-active Secret Sharing techniques which can be used to refresh the secret shares
6. have a low computation cost associated with generating new secret shares, and refreshing existing secret shares

The primary requirement for the underlying TSS is that it be cryptographically secure within the time period that the authentication mechanism is likely to be deployed.

The second of the requirements is that the TSS algorithm support a fluctuating network size. This is particularly important as the infrastructureless network environment is likely to vary in size considerably during its lifetime. A TSS algorithm that requires all secret shares to be recomputed every time the size of the network changes would be impractical.

The third of these requirements states that the TSS algorithm should be compatible with the Joint Secret Sharing techniques that permit distributed generation of secret shares. This is required as in an infrastructureless network environment the existence of trusted nodes cannot be assumed. Generation of the authentication mechanism's public and private keys as well as the secret shares must be shared amongst the nodes of the authentication mechanism.

The fourth requirement is for the TSS to support Verifiable Secret Sharing so that partial signatures on certificates can be validated. This is required to prevent malicious nodes submitting invalid partial signatures in an attempt to thwart the signature generation scheme.

Fifth in the list of requirements is that the TSS algorithm should use Proactive Secret Sharing (PSS) to periodically refresh the secret shares of valid secret share holders. This can be used to prevent a malicious party from slowly acquiring a threshold number of the secret shares.

The algorithm will likely be run on computationally limited information devices like handheld computers or mobile telephones. With this in mind, the final requirement is that the TSS algorithm be computationally efficient when initialising, generating new secret shares or refreshing of secret shares. An example of an algorithm that fulfils these requirements is that of Luo et al. [10].

5 Context Information and Authentication

In this section we provide a characterisation of security-relevant context information that can be included in the authentication process. This context must:

- be verifiable by a reliable, preferably authenticated, node
- describe some contextual attribute of the node, (e.g., location, activity)
- be sufficiently accurate, and of sufficient granularity, to have a bearing on the authentication decision

The types of context information that meet these requirements will depend on the circumstances in which the authentication mechanism is deployed. Consider the example of an emergency response team that arrives at a disaster site and seeks to setup a MANET for communication and data exchange. If the devices that will form the network are all from the same response team it is likely that they will have traditional means of authentication, such as public key certificates. To strengthen the authentication provided by these traditional authentication mechanisms, and to cope with situations where the public key used by one device cannot be verified by another device (such as when non emergency workers, who bring their own devices, arrive on site to advise the emergency response team), context authentication can be used.

Each context authentication factor supported by the authentication mechanism will be agreed upon when the authentication mechanism is started. Each of these factors will also require a value, quoted in partial signatures. The same applies to traditional authentication factors. This value is based on the increase in confidence, or certainty, the authentication mechanism has in the identity of the factor user. Factors that deliver a large increase in certainty, as might be the case with public key authentication, have a higher value than factors that deliver a marginal increase in identity certainty for a new node (e.g., proximity-based authentication).

Table 1 contains context factors that might be used in the emergency response scenario. Example values are also provided for the context factors. An assumption is made that the signature threshold for the TSS is 15 partial signatures.

Table 1. Example context authentication factors used by devices in an emergency scenario

Context Types	Value in Partial Signatures
device proximity	5
verifiable location history	10
verifiable interaction history	10

6 Context-Enhanced Authentication for Infrastructureless Environments

In this section we present our multi-factor, context-enhanced authentication mechanism for infrastructureless environments. Our mechanism requires a new node to locate authentication mechanism nodes and use factors with those authentication mechanism nodes to authenticate. For each successful use of a factor, the new node receives a number of partial signatures towards its authentication certificate. The number of partial signatures received depends on the value of the factor.

We assume that the infrastructureless network is comprised of mobile devices under the control of human users. These devices have long-range wireless communication abilities, and possibly support range-limited communication such as infra-red or physical contact-based communication. These devices are also able to obtain trusted context information either by sensing it themselves or from sources that they trust. Each of the devices is computationally capable of performing public key cryptography, and is able to establish a secure communication channel to any of the other devices in the network.

The operation of our authentication mechanism progresses in two distinct phases: the initialisation phase where the authentication mechanism is set-up, and the post-initialisation phase which begins immediately after the initialisation and continues until the infrastructureless network ceases to exist. For the purposes of this paper we concentrate on the creation and operation of the authentication mechanism only. Revocation of certificates is considered outside the scope of this work.

6.1 Initialisation Phase

The initialisation phase sees a group of nodes come together and agree to form the authentication mechanism. As it is likely these nodes are mutual strangers we make the assumption that an external channel is used to bootstrap a trust relationship between this initial group of nodes. This bootstrapping process might take the form of node users sharing secret information verbally, or possibly through physical contact of the nodes. Once a trust relationship has been established between the nodes, the operational parameters of the authentication mechanism can be decided. These parameters are recorded in the authentication mechanism policy, and include:

- the authentication factors to be supported
- the threshold signature scheme to use
- the signature threshold value, t , for the TSS algorithm
- the number of secret shares to allocate to each node in the authentication mechanism, (referred to as k)
- the value of the authentication factors

Once the operational parameters for the authentication mechanism have been decided, the initialisation phase proceeds as follows:

1. the nodes distributively generate a public and a private key for the authentication mechanism
2. each node in the initial group receives k secret shares in the authentication mechanism private key

To generate an authentication certificate, t duplicates of an unsigned version of the certificate must be partially signed by other authentication mechanism nodes. This proceeds as follows:

1. node M generates t unsigned duplicates of an authentication certificate for itself
2. M then constructs an $USE(factor)$ message and sends it to the other authentication mechanism nodes
3. node L , capable of checking authentication factor $factor$, responds with $WILL_CHECK(factor)$
4. M sends L copies of its unsigned certificates, with the number sent equal to the value of $factor$
5. L then checks that M can use $factor$
6. if the check succeeds, L partially signs the unsigned certificates provided by M using a different one of its secret shares on each of the unsigned certificates. L then returns the now partially signed certificates to M .
7. M repeats steps 2 to 6 using different factors until it has t partially signed certificates. M can then combine the partially signed certificates to produce a valid authentication certificate for itself.

In step 3 it is possible that multiple nodes reply to M 's request. The choice of which respondent to use the factor with will depend on the type of $factor$. For example if $factor$ is distance dependent, then M will likely choose the closest respondent. If there are no respondents in step 3, M must repeat step 2 with a different factor. If M cannot do this, then M cannot be authenticated and the protocol terminates.

Assuming the completion of the initialisation phase, each node in the initial group is now part of the authentication mechanism and in possession of:

- the authentication mechanism policy
- the authentication mechanism public key
- k secret shares in the private key
- an authentication certificate for itself

As M is required to interact with different authentication mechanism nodes as part of the authentication process it may be tempted to engage in “double spending” whereby it uses the same factor multiple times. Our approach to deal with this is discussed in Section 7.

6.2 Post-initialisation Phase

After the initialisation phase is completed all future authentications of new nodes proceed according to the steps below:

1. node M locates an authentication mechanism node and requests the authentication mechanism policy
2. M then generates t unsigned duplicates of an authentication certificate for itself
3. M constructs an $USE(factor)$ message and sends it to the authentication mechanism nodes, specifying the factor it wants to use as the term $factor$
4. one of the authentication mechanism nodes, L , who can check authentication factor $factor$, responds with $WILL_CHECK(factor)$

5. M then sends L a number of duplicates of its unsigned authentication certificate, with the number sent equal to the value of *factor*
6. L then checks that M can use *factor*
7. if the check succeeds, L partially signs the unsigned certificates provided by M using a different one of its secret shares on each of the certificates M provided, before returning the partially signed certificates to M
8. M must repeat steps 3 to 7 using different factors to acquire t partially signed certificates. Once this is done, M can combine the partially signed certificates to produce a valid authentication certificate for itself

M may receive duplicates of the WILL_CHECK message in step 4. These duplicates are handled as per the method described in the initialisation phase.

If M reaches a stage where it has no more factors it can use, and has not acquired the threshold number of partial signatures, then the protocol terminates and M is not authenticated.

Provided that M is able to complete step 8, it will be in possession of the following:

- the authentication mechanism public key
- the authentication mechanism policy
- a signed authentication certificate for itself

If M is permitted to become one of the authentication mechanism nodes, it requests an allocation of secret shares according to the process defined below:

1. M sends a REQUEST_JOIN message to the nearest node in the authentication mechanism along with its authentication certificate
2. the authentication mechanism nodes verify M 's authentication certificate using the authentication mechanism public key
3. provided M is permitted to join the authentication mechanism, k secret shares are generated for M and securely communicated to it

7 An Extended Version of the Authentication Scheme to Support Authentication Levels

Our approach enables new nodes to authenticate using a combination of different factors. These could be a small number of high-value factors, or a large number of low-value factors. To differentiate between these two cases, we extend the authentication mechanism to support *Authentication Levels*. To achieve a higher Authentication Level, nodes must use higher-valued factors. The number of these Authentication Levels and the associated entry requirements will be implementation specific, and thus decided during the initialisation phase of the authentication mechanism.

At this stage we have identified two possible methods for implementing Authentication Levels. In the first method, each Authentication Level is represented

by a separate TSS, with all the schemes run concurrently. In this arrangement each of the TSS could use different thresholds. The lower Authentication Level would require fewer partial signatures to achieve than do subsequent higher Authentication Levels.

The second method of implementing the Authentication Levels uses only a single TSS and thus has lower overhead than the first approach. As there is only one TSS there can be only one signature threshold and so to authenticate all nodes must acquire the same number of partial signatures. The assignment of the Authentication Levels needs to be done after the authentication certificate is generated so that the authentication factors used can be checked. This checking requires a record of the factors used to be attached to the authentication certificate. This record can also be used to detect illegal “double spending” where new nodes use the same authentication factor multiple times to gain the threshold number of partial signatures.

When examined, the computational cost of the second method is significantly less than that of the multi-TSS implementation where the costs increase linearly with the number of threshold signature schemes operated. The two methods offer a trade-off between increased computation in the case of the first method, and increased communication with the added difficulty of establishing lists of credentials, in the case of the second method. The decision as to which of the methods to use should be made during the initialisation phase and need to take into account the computational capacity of the network.

8 Evaluation

In this section we present the results of two sets of simulations we performed to examine the operation of our authentication mechanism. The simulations assumed a worst case scenario where factors, and the abilities to check factors, were spread randomly throughout the infrastructureless network. The authentication rate of new nodes that applied to join the network was measured when altering: the number of neighbouring authentication mechanism nodes, the chance of particular factors occurring in the network, and the value of factors supported by the authentication mechanism nodes.

In both sets of simulations each new node could have a maximum of 10 authentication factors (no distinction was made between context and traditional authentication factors). Each factor was assigned to a new node with probability p . Authentication mechanism nodes could check a maximum of 10 different kinds of authentication factors. The probability of an authentication mechanism node being able to check a particular factor was also defined as p .

The simulations were conducted under the assumption that the authentication of a new node failed if the new node was unable to acquire the threshold number of partial signatures. This occurred if the new node did not have enough factors to authenticate, or when the new node could not locate authentication mechanism nodes capable of checking its factors.

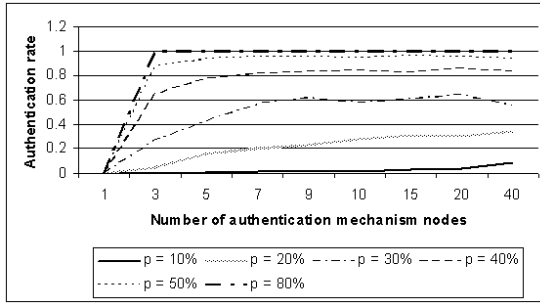


Fig. 1. Graph of authentication rate for varying numbers of authentication mechanism nodes

8.1 Simulation Set One

In the first set of simulations the effect on authentication rate of altering the number of authentication mechanism nodes was examined. Authentication mechanism node populations of 1, 3, 5, 7, 9, 10, 15, 20 and 40 were used. The chance of a new node having any one of the 10 supported factors was calculated for $p = 10\%$, 20% , 30% , 40% , 50% and 80% . All factors were given a value of 5, and the signature threshold was set at 15.

The results, plotted in Figure 1, suggest that high authentication rates can be achieved, even with small numbers of authentication mechanism nodes. This is provided that the authentication factors support by the authentication mechanism nodes occur frequently in the network (this corresponds to p being greater than 40% in our simulations).

8.2 Simulation Set 2

This set of simulations examined the relationship between the authentication rate and authentication factor value. Factor values of 1, 2, 4, 5, 6, 7, 8 and 10 were used during the testing. In addition, different simulations were performed with $n = 5, 10$ and 15 , where n refers to the number of authentication mechanism nodes the new node could contact. The value of p was held constant at 50% (i.e. $p = 50\%$) over all the tests. The signature generation threshold was set at 20.

From Figure 2 it can be seen that, for the particular parameters used in this simulation, increasing the number of authentication mechanism nodes beyond 5 had little effect on authentication rate. Far more important were the value of the factors used by new nodes. Extremely low factor values (less than 3) prohibited the authentication of new nodes, while high values (approaching 10) resulted in very high authentication rates. This can be explained as follows. For a factor value of 1, authentication was impossible. For a factor value of 2, the new nodes needed to have (and use) all 10 possible factors to reach the signature threshold of 20. Factor values of 7 and above required at most 3 factors, making authentication much more likely. As can be seen on Figure 2, the general increase in authentication rate was interrupted when the factor value equalled 6. This was

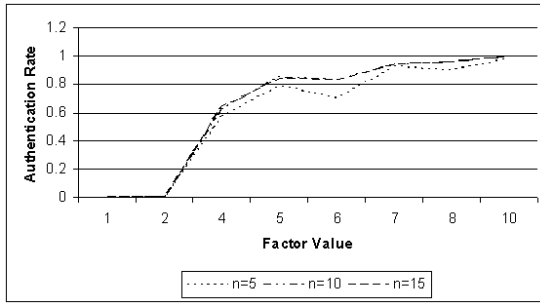


Fig. 2. Authentication rate of new nodes when varying factor value

because when the factor value was set to 5, each new node had to use 4 factors to gain the 20 partial signatures needed to authenticate. These 20 partial signatures had to come from at least 2 different authentication mechanism nodes (each authentication mechanism node had 10 secret shares). With a factor value of 6, each new node still had to use 4 factors to gain the required minimum 20 partial signatures. However, the partial signatures had to come from 4 authentication mechanism nodes (as each authentication mechanism node only had 10 secret shares, it could only check and partially sign for 1 factor). This had the effect of halving the population of authentication mechanism nodes available to the new node. As would be expected in this case, the anomaly was strongest for the plot of $n = 5$, and weakest for the plot of $n = 15$. The authentication rate recovered again for factor values of 7, as only three factors had to be used by a new node to obtain the threshold 20 partial signatures.

9 Conclusions

In this paper we presented a multi-factor, context-enhanced authentication mechanism capable of operating effectively in an infrastructureless network environment. The main contributions of the paper are that (1) it presented an authentication mechanism that makes use of both traditional and contextual information factors, (2) provides a characterisation of security-relevant contextual information, (3) describes the operation of the authentication mechanism, and (4) provides simulations to determine the effect of varying operational parameters for the authentication mechanism. These simulations were conducted for a worst case scenario where high node mobility resulted in the presence of particular factors, and the ability to check them, being randomly distributed throughout the infrastructureless network.

Based on the results of the simulations, increasing the value of factors was found to increase the authentication rate of new nodes significantly. Most importantly for infrastructureless network environments, our simulations suggest a small number of authentication mechanism nodes can authenticate new nodes with a very high success rate, provided that the authentication factors supported

by the authentication mechanism are carefully chosen to have a high probability of occurring in nodes within the infrastructureless network.

References

1. Neuman, B., Ts'o, T.: Kerberos: An Authentication Service for Computer Networks. *IEEE Communications* **32**(9) (1994) 33–38
2. Shamir, A.: How to Share a Secret. *Communications of the ACM* **22**(11) (1989) 612–613
3. Luo, H., Kong, J., Zeros, P., Lu, S., Zhang, L.: Self-securing Ad Hoc Wireless Networks. In: *Proceedings of the Seventh International Symposium on Computers and Communications, ISCC 2002*. (2002) 567–574
4. Ingemarsson, I., Simmons, G.: A Protocol to Set Up Shared Secret Schemes Without the Assistance of a Mutually Trusted Third Party. In: *Advances in Cryptology - EUROCRYPT'91. Lecture Notes in Computer Science*, Springer-Verlag (1991) 266–282
5. Feldman, P.: A Practical Scheme for Non-Interactive Verifiable Secret Sharing. In: *Proceedings of the 28th Annual Symposium on the Foundations of Computer Science, IEEE* (1987) 427–437
6. Hertzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: *Proceedings of CRYPTO'1995*. (1995) 339–352
7. Pirzada, A., McDonald, C.: Kerberos Assisted Authentication in Mobile Ad Hoc Networks. In: *27th Australasian Computer Science Conference*. (2004)
8. Zhou, L., Haas, Z.: Securing ad hoc networks. *IEEE Networks* **13**(6) (1999) 24–30
9. Saxena, N., Tsudik, G., Yi, J.: Efficient Node Admission for Short-lived Mobile Ad Hoc Networks. In: *IEEE Conference on Networking Protocols (ICNP)*. (2005)
10. Luo, H., Kong, J., Zeros, P., Lu, S., Zhang, L.: URSA: ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Transactions on Networking* **12**(6) (2004) 1049–1063
11. Glynos, D., Kotzanikolaou, P., Douligeris, C.: Preventing Impersonation Attacks in MANET with Multi-Factor Authentication. In: *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05)*. (2005) 59–64
12. Covington, M.: A Flexible Security Architecture for Pervasive Computing Environments. PhD thesis, College of Computing, Georgia Institute of Technology (2004)