Managed authentication

David Irvine, david.irvine@maidsafe.net, maidsafe.net limited (registered in Scotland Sc 297540)

September, 2010 ***Abstract*—Today all known access mechanisms that grant access to distributed or shared services requires a server or authoritative control in some form. This presents many issues, including security, trust and privacy to name only a few. This paper presents a system of authentication that abolishes the requirements for any user-name or password containers as lists or similar. It also negates the necessity for any server based systems as a login entity for people to connect with prior to gaining access to a system for any reason.**

***Index Terms*—security, freedom, privacy, authentication**

Contents

List of Tables

List of Algorithms

I. Introduction

AUTHENTICATION allows access to a system at a certain level or privilege. This is described in *self authentication* [1]with reference to an identity created, managed and maintained in seclusion, or in another way of thinking, anonomously. In many other situations this is not a situation that is beneficial, in some situations control over access in large groups is required, such situations would be corporate networks, military and many others.

A. Conventions Used

There is scope for confusion when using the term "key", as sometimes it refers to a cryptographic key, and at other times it is in respect to the key of a DHT "key, value" pair. In order to avoid confusion, cryptographic private and public keys will be referred to as K_{priv} and K_{pub} respectively, and DHT keys simply as keys.

- Node $\equiv$ a network resource which is a process, sometimes referred to as a vault in other papers. This is the computer program that maintains the network and on its own is not very special. It is in collaboration that this Node becomes part of a very complex, sophisticated and efficient network.
- H $\equiv$ Hash function such as SHA, MD5, etc.
- $PBKDF2_{[Passphrase][Salt][IterCount]} \equiv$ Password-Based Key Derivation Function or similar
- XXX_{priv}, $XXX_{pub} \equiv$ Private and public keys respectively of cryptographic key pair named XXX
- $AsymEnc_{[K_{pub}]}(Data) \equiv$ Asymmetrically encrypt Data using K_{pub}
- $AsymDec_{[K_{priv}]}(Data) \equiv$ Asymmetrically decrypt Data using K_{priv}
- $Sig_{[K_{priv}]}(Data) \equiv$ Create asymmetric signature of Data using K_{priv}
- $+ \equiv$ Concatenation
- $STORE \equiv$ Network or other key addressable storage system

II. Outline

A. Creating a user access key

1) USER creates an ID as described in *self encryption* [1] or has some other mechanism to send and receive secure communications
2) USER requests permission from manager to join XXX network, via an encrypted message (signed) to $MANAGER@XXX$
3) Manager creates $USER_{pub} \& USER_{priv}$
4) $H(USER_{pub} + Sig_{MANAGER_{prv}})$ is stored on $STORE$, this also will be the access ID for USER
5) Manager creates a $H(USER@XXX)$ packet in $STORE$ (this is the contact address of USER when connected with company XXX)
6) MANAGER sends $USER_{priv}$ to USER via an encrypted message.

B. Accessing the network

1) MANAGER creates a share as described in *maidsafe distributed file system* [6] if none already exist
2) The USER ID $H(USER_{pub} + Sig_{MANAGER_{prv}})$ is added to the list of ID's authorised access identities
3)

C. Banning user from network

As the USER identity packet is signed by the MANAGER, then the manager can delete this packet. This renders the USER access key useless and all access revoked. To ensure

this is absolute, the message type sent cannot be used to verify an ID (by sending pub plus signature in a packet for hashing be the recipient) and insted a $STORE$ lookup for a valid has to be happen every time.

D. Boostrapping a network

1) N plus P key sharing: As the initial node has an extreme amount of power and no equal to absorb this power should that identity become unstable for whatever reason (such as retiring, leaving or simply becoming hostile) then a mechanism must be put into place to ensure this ID is not stand alone.

III. Conclusions

T

References

[1] David Irvine, self authentication, david.irvine@maidsafe.net
[2] David Irvine, Self Encrypting Data, david.irvine@maidsafe.net
[3] David Irvine, peer to peer public key infrastructure, david.irvine@maidsafe.net
[4] David Irvine, maidsafe.net, a new network paradigm , david.irvine@maidsafe.net
[5] David Irvine, Autonomous network, david.irvine@maidsafe.net
[6] David Irvine, maidsafe distributed file system, david.irvine@maidsafe.net

David Irvine is a Scottish Engineer and innovator who has spent the last 12 years researching ways to make computers function in a more efficient manner.

He is an Inventor listed on more than 20 patent submissions and was Designer / Project Manager of one of the World's largest private networks (Saudi Aramco, over $300M). He is an experienced Project Manager and has been involved in start up businesses since 1995 and has provided business consultancy to corporates and SMEs in many sectors.

He has presented technology at Google (Seattle), British Computer Society (Christmas Lecture) and many others.

He has spent many years as a lifeboat Helmsman and is a keen sailor when time permits.