Authentication in Ad hoc Networking

A O Salako

University College London

Abstract: Authentication is an aspect of communication network security that deals with ensuring that the principals with whom one interacts are the expected ones. Informally, authentication allows the receiver to verify that the claimed sender really sent the data. Authentication inherently secures other aspects of ad hoc wireless communication such as freshness, availability, integrity and confidentiality.

The definition of Ad hoc Networking continues to evolve broader from its literary meaning - Network for a particular purpose. The scenario, envisaged in the recent past, has become an immediately applicable, if not lagging behind. But our greatest fear is deploying this next generation technology without the absolutely necessary security measures put in place.

This paper proposes an authentication protocol for the most basic ad hoc peer-to-peer homogeneous device communication. This simple lightweight authentication protocol is a based on Tiny Encryption Algorithm, TEA.

1. Introduction

Ad hoc Networking is could be described as an evolution from Mobile IP (The convergence of mobile and data technology) [1,2], through Packet Radio Networks [3]. Further research is ongoing on the advancement to MANET (Mobile Ad hoc Networking) [3] and Ubiquitous Computing [4]. The English dictionary defines ad hoc as an adjective meaning "Formed for or concerned with one specific purpose; Improvised and often impromptu".

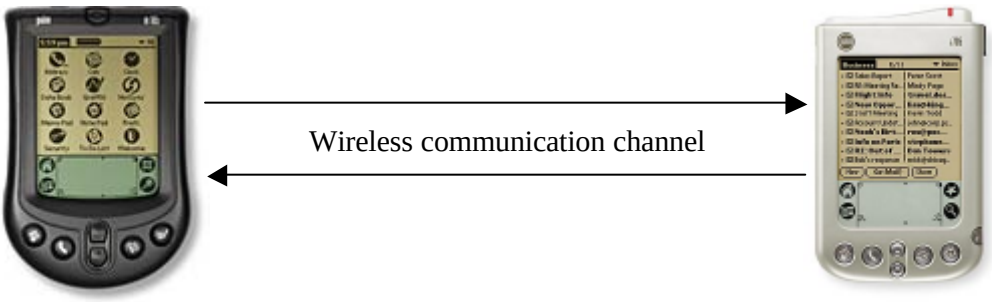
Ad hoc networking enables wireless devices to network with on another, as needed, even when access to the Internet is unavailable. It enables a wide range of powerful applications, from instant conferencing between notebook PC users to emergency and military services that must perform in the harshest conditions. Wireless communications without routers, base stations, or Internet Service Providers. An ad-hoc network might consist of several home-computing devices, plus a notebook computer that must exist on home and office networks without extra administrative work. Key applications - conferencing, home networking, emergency services, Personal Area Networks, *Bluetooth*, and more. Addressing the key challenges of ad hoc networking - resource management, scalability, and especially security is of the essence [3].

Wireless communication, such as in ad hoc networking, can be highly vulnerable to security threats. Authentication is a security primitive which enables a node to ensure the identity of the peer node it is communicating with. In most applications where security matters, authenticity is an essential prerequisite. Granting resources to, obeying an order from, or sending confidential information to a principal of whose identity we are unsure is not the best strategy for protecting availability, integrity and confidentiality. The ad hoc network environment introduces an advanced security problem: the absence of an online server. When a node comes within range, we cannot connect to an authentication server to obtain a Kerberos ticket or to check the validity of an exhibited certificate: suddenly, the traditional solutions for wired networks no longer apply [5].

On a microscopic scale, SPINS (UC Berkeley) [6] authenticate peer-to-peer communication via a third party (base station), which is arguably antithetical to ad hoc networking, if not outright contradictory. The Resurrecting Duckling policy [5] suggests giving peers credentials that allow them mutually authenticate each other. This research is an attempt to use Tiny Authentication Algorithm, TEA to authenticate the most basic ad hoc peer-to-peer communication e.g. two PDAs for a practical wireless communication session.

2. Use Case

In the common use case, where two devices are to be used in the most basic ad hoc set-up as suggested in the section above, there is usually human presence, which intervenes like a base station. This practical assumption of human presence, at least at initiation, is in line with this basic definition of ad hoc networking and the use case envisaged.



Ad hoc Network

Frank Stajano [5] suggests a means of authenticating peer-to-peer interaction in this scenario. The human user (master) allows one to perform the special action of uploading a new policy in a node (peer); but, apart from that, any action can be invoked by any principal who presents the required credentials, as required by the node's then current policy. A human user will be master to the nodes (probably via a cyber-intermediary) and will give them the credentials that allow them to talk to each other – credentials that the nodes not participating in this particular session won't have, even if they come from the same manufacturer. For every possible action, security rules for the node would state which credentials the principal should exhibit in order to make the node carry out the action. The problem that arises is that at the bootstrapping base, a principal can acquire the imprinting key of a node, and can consequently take control of the node. A multilevel integrity system controls this problem. The various parts of the policy would be ranked at different integrity levels, so that one could allow the low integrity items to be rewritten but not the high integrity ones, which would include the most sensitive actions such as taking over.

Part of the SPINS [6] work, SNEP (Secure Network Encryption Protocol) provides two-party data authentication as one of the baseline security primitives. This is done via a third party (base station), which each node trusts, because of the extremely resource constrained sensor nodes in their case. This prevents computationally expensive public-key cryptography protocols for symmetric-key setup.

A receiver should be able to ascertain the origin of a message. Also in this authentication process, the deceitful intruder should be detected [8]. Ross Anderson buttresses this point by distinguishing authentication from safety:

> "Security involves making sure things work, not in the presence of random faults, but in the face of an intelligent and malicious adversary" [9].

3. Research

SPINS used RC5 Encryption Algorithm due to severely limited code size. Algorithms such as TEA or TREYFER are smaller alternatives, but RC5 was chosen because the security of these other ciphers is not yet thoroughly analysed. This research, comparable to the SPINS work, is to design a smaller and more secure or alternative authentication algorithm/protocol for such resource-constrained devices (sensor networks) based on TEA - Tiny Encryption Algorithm.

TEA is one of the fastest and most efficient cryptographic algorithms in existence. It was developed by David Wheeler and Roger Needham at the Computer Laboratory of Cambridge University [7]. It is a *Feistel* symmetric block cipher, which uses operations from mixed (orthogonal) algebraic groups - XORs and additions in this case. It encrypts 64 data bits at a time using a 128-bit key. TEA Extension addresses a couple of minor weaknesses (irrelevant in almost all real-world applications), and introduces a block variant of the algorithm that can be even faster in some circumstances. It is a program that will run on most machines and encipher safely. It uses a large number of iterations rather than a complicated program. It is hoped that it can easily be translated into most languages in a compatible way. The first program uses little set up time and does a weak non linear iteration enough rounds to make it secure. There are no preset tables or long set up times. It assumes 32 bit words.

It is a *Feistel* symmetric block cipher though addition and subtraction are used as the reversible operators rather than XOR. The routine relies on the alternate use of XOR and ADD to provide nonlinearity. Using a 128-bit master key, *K[0..3]*, and a simple key schedule: odd rounds use *K[0,1*] as the round subkey, and even rounds use *K[2,3]*. Two rounds of TEA applied to the block Y_i, Z_i consists of:

$$c = c + d \qquad Y_i + 1 = Y_i + F(Z_i, K[0, 1], c) \qquad Z_i + 1 = Z_i + F(Y_i + 1, K[2, 3], c)$$

where the round function ***F*** is defined by

$$F(z, K[i, j], c) = (SL_4(z) + K[i]) \text{ XOR } (z + c) \text{ XOR } (SR_5(z) + K[j])$$

Here $SL_4(z)$ denotes the result of shifting (not rotating) z to the left by 4 bits, and $SR_5(z)$ denotes a shift to the right. In this description, c is a value which perturbs the F function so that it is different in each round. Before each cycle, c is incremented by a fixed constant $d = [(v5 - 1)2^{31}]$; c is initially 0.

Encoding Cipher (Cryptographic Algorithm) Routine in C:
Routine, written in the C language, for encoding with key *k[0] - k[3]*. Data in *v[0]* and *v[1]*.

```
void code(long* v, long* k){
unsigned long y=v[0],z=v[1], sum=0,         /* set up */
     delta=0x9e3779b9,                      /*    a    key    schedule
constant */
     n=32;
while (n-->0)   {                    /* basic cycle start */
     sum += delta ;
          y += (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1];
          z += (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3];     /*  end  cycle
*/
          }
v[0]=y ; v[1]=z ;              }
```

NB: This might afford a chance for a documented cryptanalyses of TEA [8].

4. Conclusion

The work to be done now is to design a protocol that implements an authentication algorithm for the use case in the scenario described. This should be computationally inexpensive public-key cryptography protocol for symmetric-key setup. TEA, using C programming language, will be written for palm development tools and GCC with Palm OS as the implementation platform. This particular research project will ideally involve both computer based modelling (e.g. simulation on an emulator) and physical implementation (e.g. on actual PDAs) of device ensembles and their coordination algorithms.

5. References

[1] Mark Norris, *Mobile IP Technology for M-business*, 2001, Artech House; ISBN: 1580533019

[2] James D. Solomon, *Mobile IP the Internet Unplugged*, 1998, Prentice Hall; ISBN: 0138562466

[3] Charles E. Perkins, *Ad Hoc Networking*, 2001, Addison-Wesley, London; ISBN: 0-201-30976-9

[4] Frank Stajano, *Security for Ubiquitous Computing*, 2002, John Wiley and Sons, Ltd, West Sussex, England, ISBN: 0-470-84493-0

[5] Frank Stajano, *The resurrecting duckling: What Next?*, 2000.
http://www-lce.eng.cam.ac.uk/~fms27/papers/duckling-what-next.pdf

[6] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J.D. Tygar. *SPINS: Security Protocols for Sensor Networks*. Mobicom 2001
http://www.millennium.berkeley.edu/tinyos

[7] David J. Wheeler and Roger M. Needham, *TEA, a Tiny Encryption Algorithm*, 1995 and *TEA Extension*, 1996, Computer Laboratory Cambridge University
http://www.ftp.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html
http://www.cl.cam.ac.uk/ftp/users/djw3/xtea.ps

[8] Bruce Schnier, *Applied Cryptography: protocols, algorithms and source code in C*, 2ed, 1996, Wiley, ISBN: 0-471-11709-9

[9] Ross Anderson, *Security Engineering - a Guide to Building Dependable Distributed Systems*, 2001, John Willey & Sons, Inc., New York; ISBN: 0-471-38922-6

Adedamola Salako (EPSRC-funded MRes Telecommunications, University College London)
<mt01002@ee.ucl.ac.uk>

Industrial Affiliation: Prof. Ian W. Marshall (Supervisor, BT Exact)