AIM Encrypt: A Case Study of the Dangers of Cryptographic Urban Legends

Michael E. Locasto
{*locasto@cs.columbia.edu* }
Department of Computer Science
Columbia University

November 26, 2003

Abstract

Like e–mail, instant messaging (IM) has become an integral part of life in a networked society. Until recently, IM software has been lax about providing confidentiality and integrity of these conversations.

With the introduction of AOL's version 5.2.3211 of the AIM client, users can optionally encrypt and protect the integrity of their conversation. Taking advantage of the encryption capabilities of the AIM client requires that signed certificates for both parties be available. AIM (through VeriSign) makes such certificates available for purchase. However, in a "public service" effort to defray the cost of purchasing personal certificates to protect IM conversations, a website (www.aimencrypt.com) is offering a certificate free of cost for download. Unfortunately, the provided certificate is the same for everyone; this mistake reveals the dangers of a public undereducated about computer security, especially public key cryptography.

1 Introduction

IM conversations are normally sent in the clear, and the lack of encryption makes it quite easy to intercept or observe the content of a conversation. AOL has recently introduced the ability for the AIM client to use end–to–end encryption via public key cryptography [5]. This decision works well with AOL's IM system architecture.

AOL provides a number of servers that effectively proxy chat between users; that is, users do not normally directly connect or engage in a peer–to–peer session. Rather, users' chat clients connect to the AOL servers and instruct the servers to contact buddies on their behalf[1].

[1] Direct connection was recently added to AIM clients but remains problematic because many clients are either home users behind a NAT device or are employees behind a firewall.

AOL has decided to enable their clients with mechanisms for encrypting chat content and has proposed a system for both private and corporate use of their chat client. Protecting the confidentiality of both casual and mission–critical chat content is an important goal from both a privacy and economic standpoint. AOL hopes to profit by offering corporate customers the ability to encrypt and integrity protect employee communication. However, AOL plans to provide more complete service and support features to corporate clients. Home users are required to purchase or otherwise obtain a certificate in order to take advantage of the new security mechanisms.

This report focuses on one such certificate distribution mechanism that is unaffiliated with AOL or Verisign. This particular method of certificate distribution is troubling because it offers the user some hope of security while providing none at all.

1.1 AOL Instant Messenger

AOL Instant Messenger (AIM) is a popular (if not the runaway favorite) IM software. The AIM architecture is based on a client–server model with AOL's servers proxying connections for client 1–to–1 chats and chatrooms.

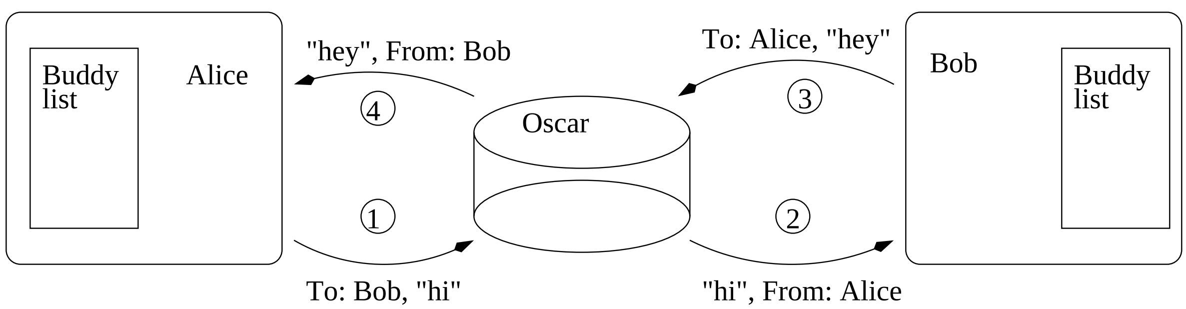

Figure 1: The general AIM architecture. AOL servers proxy connections between chat clients. To contact a buddy, a client will send a chat request to the AOL server Oscar containing the name of the buddy to initiate contact with. Oscar will then contact the named buddy and set up a bi–directional communications channel.

All this information (including login passwords) is sent cleartext, although passwords undergo a reversible transformation. When AOL considered adding encryption and integrity protection (mainly for corporate clients) the decision was made to reject an SSL-based solution. Such a solution would require the AIM servers to perform expensive cryptographic operations on all messages passing through the servers. This solution clearly places a heavy load on the AIM servers and is not practical for AIM's business model. Instead, AOL chose to require that encryption be performed end–to–end with public-key cryptography; if one entity wishes to protect their chat, both entities must employ cryptography and trade certificates [3].

The entities themselves are therefore responsible for performing encryption; the AOL servers merely retain their proxy role of passing traffic between buddies. There are several protocols that provide confidentiality and integrity in the presence of public–key cryptography primitives. Two good examples are JFK [1] and SSH-2 [4]. There is no reason to suppose that AOL's engineers actually got this complex task correct or based their protocol on one of the aforementioned well–established and well–studied protocols. However, even a robust protocol is useless if everyone uses the same public–key identity.

1.2 Public–Key Certificates

A public key certificate is a bit like a real–world social security card; it binds a number to a name. Confidence in the integrity of this binding is possible because the certificate is *signed*. Signing a piece of information provides the ability to check if the information content has been tampered with.

The content is reduced to a single very large number via a one–way transformation (like a strong hash function), and then that number is encrypted (e.g., signed) with the private key that corresponds to the public key contained in the certificate. Since all users have free access to or can be made aware of the public key via the certificate, and only the private key corresponding to the public key could have created the signature, verifying the signature is a simple matter of decrypting the signature with the public key and comparing it with the hash of the received content.

1.3 Background on Public Key Encryption

If a user Albert wishes to talk secretly with Becky, he can use Becky's public key (which everyone knows) to transform the data (encrypt it) in such a way so that it can only be decrypted with Becky's private key. Becky *does not* distribute her private key. Now, Becky can use Alberts public key to encrypt any message she wants to send to him, and only Albert can decrypt it, because only Albert has his private key. This mechanism works fine until everyone has the same public key, as proposed by AIM Encrypt.

This requirement logically means that everyone has the same private key. Thus anyone can decrypt anyone else's conversations because everyone is pretending to be the same person. All users know the only secret piece of information in the system.

2 Discussion

The AIM Encrypt website[2] is either an example of poorly executed good intentions or malicious social engineering. Even worse, this website has received a fair amount of attention after having been publicized on TechTV's *Call for Help*. It remains one of the first sites that Google returns for searches involving AIM and encryption.

Particularly troubling is the statement in the website FAQ: "There are two security keys in this certificate one is posted on this site and would normally be different for every person, thus removing one layer of security. [sic]" [2]

This statement demonstrates a clear lack of understanding about the nature of public key cryptography. A public–private keypair is unique to each user. *Both* keys are normally different for each entity, and revealing the private key removes the *only* layer of security. It may be fitting that this certificate expires on an April Fool's Day.

Each public key has an associated private key; while the public key can and should be widely distributed, the private key must be protected. What the AIM Encrypt website ensures is that everyone who uses this certificate is the same identity. Therefore, a client has no guarentee that the person she is talking to is her AIM buddy – she only knows that the buddy is this now ubiquitous "AIMEncrypt" buddy: the same buddy, ironically enough, that she is pretending to be.

If the AIM protocol calculates shared session keys and these keys are passed to the other party (as is common with hybrid encryption[3]), the protocol is trivially open to both active and passive attacks on data confidentiality and integrity.

If instead the protocol relies on Diffie-Hellman, users are still exposed to a Man–in–the–Middle attack at least the first time certificates are exchanged. Exact knowledge of the nature and extent of the threat cannot be elucidated because we are not privy to the protocol that AOL implemented.

2.1 Using the Java 'keytool' to Generate A Certificate

AOL has partnered with Verisign to sell certificates to users wishing to have confidential, integrity–protected IM. AOL is treating this as a business proposition; the typical casual end–user of AIM can obtain a certificate, but it is not worth the cost. The cost is only justified because the certificate is signed by a well–known CA.

However, for casual chatting purposes, a self–generated, self–signed certificate should suffice, especially if the certificate is distributed to AIM buddies out

[2]www.aimencrypt.com

[3]Hybrid encryption is a mechanism for getting the performance benefits of symmetric key cryptography with the security properties of asymmetric, or public-key, cryptography. Usually, plaintext is enciphered with a symmetric key cipher (e.g., AES), the key used to encipher the plaintext is encrypted with the public key of the target, and then both the encrypted key and ciphertext are transmitted to the target.

of band. There are a few different methods of generating your own certificate that cost absolutely nothing[4]

Creating a public/private keypair is quite simple:

```
[michael@fae .aimcert]$ keytool -genkey -v -alias MyScreenName -keyalg RSA \
 -keysize 1024 -dname cn=MyScreenName -validity 365 \
 -keystore /home/michael/.aimcert/aimkeystore
```

Exporting a certificate from the keystore is quite simple. The following command line creates a self-signed certificate:

```
 michael@fae$ keytool -export -v -alias MyScreenName \
 -file myscreenname.cert -keystore /home/michael/.aimcert/aimkeystore
```

```
[michael@fae .aimcert]\$ keytool -printcert -file myscreenname.cert -v
Owner: CN=MyScreenName
Issuer: CN=MyScreenName
Serial number: 3fb525a3
Valid from: Fri Nov 14 13:57:39 EST 2003 until: Sat Nov 13 13:57:39 EST 2004
Certificate fingerprints:
         MD5:  CC:B9:08:46:C0:31:9E:B7:A7:48:9D:BE:5B:1E:E2:8F
         SHA1: 98:64:9B:A2:E6:B3:AC:7C:79:08:83:D9:99:8F:43:79:2A:A4:0B:9A
```

2.2 Using the OpenSSL Tool to Generate A Certificate

It is also possible to create your own certificate with the openssl tool. Most Linux and Unix distributions come with some version of OpenSSL installed, but users may download a newer version from the openssl.org website. For Windows users or anyone unwilling to compile their own version of OpenSSL, the Java SDK may be the easier choice.

2.3 The AIM Encrypt Certificate

The certificate offered by AIM Encrypt is a PEM or DER encoded X.509 self–signed certificate for the entity cn='AIMEncrypt.com',c='US', with a serial number of zero. The certificate is valid from Monday, July 7^{th} 2003 until Saturday, April 1^{st} 2006.

The MD5 digest of the certificate is:

```
36:AB:38:71:CB:A1:8F:AC:F7:7D:B3:22:06:DD:B4:0E
```

and the signature contents are:

```
Signature Algorithm: md5WithRSAEncryption
Signature Contents:
C0:15:4B:6B:4D:E3:82:5B:27:65:EB:1A:C3:06:79:7E:78:95:4B:0D
```

[4] You don't even have to give away an email address.

```
66:4E:FE:42:A3:FC:20:05:A9:D6:81:C5:E0:6B:74:B2:79:33:21:CA
F9:A3:DD:CB:44:56:D5:D1:E1:32:9B:D4:9C:DE:D0:05:B7:0F:8C:15
29:A4:61:79:2A:0E:6C:1F:25:63:CC:7F:57:75:62:90:8E:6C:F7:F4
D0:86:09:F9:27:F9:D3:65:99:27:BA:28:BD:91:A6:63:62:4A:69:DD
CB:C5:DC:4F:D8:74:89:AB:35:73:09:33:98:04:32:B9:A7:7B:DD:42
03:FA:79:36:80:E0:74:5C
```

The public key contained in the certificate is:

```
Key type: RSA (1024 bit)
Modulus:
C7:25:09:93:58:EE:1C:C1:4C:12:C3:94:08:66:81:D6:07:6A:9B:FE
1A:FC:88:96:D0:A3:44:CA:A1:1A:FC:FD:A3:47:37:1E:89:0C:4A:58
25:BE:49:48:4A:91:E6:93:47:2A:A2:AD:9B:E0:0F:19:1E:BF:7E:5C
0C:5C:A9:D7:05:AF:55:7C:9E:95:90:50:7A:8C:E2:B1:B5:EB:52:FE
21:41:85:F7:A5:46:D6:6E:48:88:FE:E6:72:D4:CF:40:9A:1E:A0:7B
CD:23:42:E1:97:22:17:A9:71:FC:2B:69:E5:B7:93:C8:D8:5E:B5:3F
1E:BA:B9:21:5F:C5:BA:01
Exponent: 0x010001
```

2.4 The AIM Encrypt Private Key

The private key is protected by the passphrase “g8dJ82kjfjq32h” and is stored in a certificate with a PKCS 12 file format, again for the entity cn=’AIMEncrypt.com’ c=’US’.

The certificate containing the private key has the following signature:

```
Signature Algorithm: md5WithRSAEncryption
Signature Contents:
57:18:89:CE:35:2C:1B:08:64:EB:36:41:E6:90:8C:11:38:86:3F:9A
B1:65:9F:0D:B0:DA:4B:A2:E4:11:5E:4F:75:F2:69:99:B3:99:E8:19
AA:82:F6:B4:B4:02:19:58:B7:AA:18:AC:76:16:4A:8E:35:68:11:C8
22:34:C7:3A:76:C7:61:95:11:7D:D1:FC:81:A9:A9:65:EE:1B:63:BD
6D:AA:99:39:AB:D7:34:0B:B5:06:DC:7B:8A:78:30:F0:1E:28:39:38
56:9B:CE:36:A2:E7:4E:19:D3:00:B9:CF:53:74:C0:2A:FE:EE:0A:7B
BC:85:CE:AA:9D:FB:A6:0F
```

which verifies the MD5 digest:

```
15:8F:56:2D:B3:C3:EE:C3:91:25:DB:3F:73:A3:66:37
```

The private key itself is:

```
Key type: RSA (1024 bit)
Modulus:
A7:A1:D6:F1:3C:98:95:72:30:CD:5F:C1:4A:9D:3D:A2:9A:AA:5A:F4
C4:CD:43:44:20:0C:31:91:D8:57:40:76:81:53:0C:CA:40:65:94:18
8C:04:B0:9B:F8:71:1A:4B:3E:7D:D3:C5:EB:9F:BC:76:7F:FA:7E:8C
```

```
39:08:5E:A0:56:A3:F7:27:B2:36:B4:63:70:AA:27:B8:AE:15:6D:3D
A7:5B:C2:EF:BE:D1:E3:5A:EE:AB:45:5F:10:9D:1B:23:CC:A7:30:79
13:29:F7:13:6D:F4:10:CE:CD:08:B8:EB:89:F9:32:81:07:77:5A:C7
5B:BC:8A:65:9D:32:17:A9
Exponent: 0x010001
```

3 Conclusion

This paper has presented a brief case study of the dangers inherent in having a public with little or no understanding of network security or basic cryptographic concepts. As security is bolted onto existing applications and infrastructures, both users and system designers should be aware of how security mechanisms should be employed.

As Bruce Schneier often comments, "security is a process, not a product." Poorly used security techniques, especially cryptographic techniques, are often a greater liability to use than not. In the simplest case, if encryption is used in such a way that voids the promise of confidentiality or integrity, then the user is worse off than if they had not used encryption to begin with. First, they have wasted machine resources and their time performing useless cryptographic operations. Second, they now have a false sense of security and are perhaps more vulnerable in this mindset.

It remains one of the critical tasks in computer science and information technology to educate both the public as well as current and future software engineers about the strengths and weaknesses of different security measures.

References

[1] W. Aiello, S. M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A. D. Keromytis, and O. Reingold. Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols. In *Proceedings of the 9th ACM International Conference on Computer and Communications Security (CCS)*, pages 48–58, November 2002.

[2] AIMEncrypt. Aimencrypt. In *www.aimencrypt.com*, 2003.

[3] AOL. About aim personal certificates. In *enterprise.netscape.com/products/aim/personalcerts/*.

[4] Daniel J. Barrett and Richard Silverman. *SSH: The Secure Shell, The Definitive Guide*. O'Reilly, 2001.

[5] Ferris Research. AOL Enterprise AIM Services Adopts PKI-based Security Over SSL, July 2003.