

AG Security

Access Grid Retreat

April 15, 2003

Robert Olson



Retreat 2003

What does security mean?

- No one can hear our budget discussion
- I can tell exactly who is hearing our budget discussion
- I won't get fired if I use the Secure Access Grid
- I won't get fired, and I can blame Bob and Terry and Lisa if someone breaks in while I'm using the Secure Access Grid
- I can put my AG Node behind the department firewall and everything will be cool
- Everything is encrypted and password protected
- The script kiddies won't get me



Retreat 2003

User Expectations

- Privacy
 - Media streams
 - Shared data
 - Presentation material
- “Locked room” analog
 - Access control for sessions
- Presence
 - Who is in the room with us?
 - Is Jim connected in this meeting?
- Single Sign-on



Retreat 2003

Key Concepts

- Identification
- Authentication
 - Verification of identity
- Authorization
- Confidentiality and Integrity

Where does Cryptography fit in?

- What is cryptography?
 - Mathematical functions used for encryption and decryption
- One of many building blocks underlying secure systems
- Schneier's two types of cryptography:
 - One stops your kid sister from reading your files
 - The other stops major governments from reading your files
- We're shooting for the latter, too



Retreat 2003

However...

Most systems are not designed and implemented in concert with cryptographers, but by engineers who thought of cryptography as just another component. It's not. You can't make systems secure by tacking on cryptography as an afterthought. You have to know what you are doing every step of the way, from conception through installation.

<http://www.counterpane.com/whycrypto.html>



Retreat 2003

Enough Crypto Theory to Get You By

- Three basic building blocks
- *Encryption* provides confidentiality, can provide authentication and integrity protection
- *Checksums and hash algorithms* provide integrity protection, can provide authentication
- *Digital Signatures* provide authentication, integrity

Shared-key (Convention) Encryption

- Sender and receiver *share* the same key
- Message is *encrypted* using that key, transferred over an insecure channel, and *decrypted* using that same key.
- Problem: How to securely transfer the key from sender to receiver
- RC4, Blowfish, AES, DES, 3DES, IDEA

Public-key Encryption

- Sender and receiver used matched *private* and *public* key pairs.
- Anyone can encrypt with a public key, but only the holder of the private key can decrypt.
- Only the holder of the private key can encrypt with that key, but anyone can decrypt with it.
- RSA, DH, Elgamal

Hash Function

- Computes a unique fixed-size “fingerprint” for a message
- One-way function: difficult to create a message with the same hash value as another message
- Hence, forgery difficult
- However, any alteration of data creates a different hash
- MD5, SHA-1, RIPEMD-160

Message Authentication Code (MAC)

- Adds a password to the hash
- Only the password holder can generate the MAC for a given document and password
- HMAC-MD5, HMAC-SHA

Digital Signature

- Encryption of a message hash with as sender's private key
- Allows a receiver to verify the authenticity and provenance of a digitally signed message

Combining this tech for real-life use: Certificates

- Public-key Certificate:
 - Someone's *public key*,
 - *Signed* by a trustworthy party
- Given a certificate, one can
 - Verify that a message claiming to have originated somewhere actually did
 - Encrypt messages so that they can only be decrypted by the owner of the certificate

What do we Identify?

- Users
 - Each user has an identity certificate
- Services
 - Authentication is a two-way street
 - Each securely-accessed service also has an identity certificate

Where is the private key?

- Q: If the public key is in the cert, where is the matching private key?
- A: The user must keep track of it.
 - Usually on his computer
 - Proxying systems keep a copy online somewhere

X.509 Certificates

- ISO standard for a public key certificate format
- Each user (*subject*) has a distinct name
- Issued by trusted *Certification Authority (CA)*

Version
Serial Number
Algorithm Identifier
Issuer name
Period of validity
Subject name
Subject's public key
Signature

But...

Whatever you do, stay away from X.509 certificates.

Ferguson and Schneier, **Practical Cryptography.**



Retreat 2003

An aside: Naming

- One of the harder problems in PKI deployment
- X.509 certificates include the names of the subject (holder of certificate) and issuer of the certificate.
- Names are "X.500 Relative Distinguished Names"
- Relic from the drive for universal directory system
- Me (in one incarnation:
 - O=Grid, O=Globus, OU=mcs.anl.gov, CN=Bob Olson



Retreat 2003 Hierarchical naming space

How to choose?

- Public CAs typically set C=CA country, O=CA name, OU=certificate type, CN=user name
- Private CAs (mostly people or organisations signing their own certs) typically set any DN fields supported by their software to whatever makes sense for them

Examples

- DOE Science Grid
 - CN= John K. Doe 1W2D3;
OU=People; O= doegrids.org
 - CN= John K. Doe 1W2D3;
OU=People; DC= doegrids; DC=org
- Globus Test CA
 - O=Grid, O=Globus, OU=mcs.anl.gov,
CN=Bob Olson
- NCSA Alliance CA
 - /C=US/O=National Computational Science
Alliance/CN=John Doe



Retreat 2003

Identity Certificates

- For example, a Globus identity certificate:

```
% openssl x509 -noout -text -in ~/.globus/usercert.pem
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 6060 (0x17ac)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=US, O=Globus, CN=Globus Certification Authority

Validity

Not Before: Jan 7 20:22:19 2002 GMT

Subject information

Not After : Jan 7 20:22:19 2003 GMT

Subject: O=Grid, O=Globus, OU=mcs.anl.gov, CN=Bob Olson

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Subject Public Key

Modulus (1024 bit):

00:cd:7d:bb:ae:30:bb:c1:74:2d:e4:6e:d4:30:6e: [etc]

Exponent: 65537 (0x10001)

X509v3 extensions:

Netscape Cert Type:

SSL Client, SSL Server

Signature Algorithm: md5WithRSAEncryption

CA Signature

23:14:96:05:0d:db:ce:aa:70:17:03:5a:07:31:a0:81:e3:10:

ACCESS GRID

Retreat 2003

How are Certificates Created?

1. Creates a private key / public key pair
2. Save the private key somewhere safe.
3. Create a *certificate request* document containing some user information and the public key
4. Submit the request to a CA.
5. CA signs the request
6. User retrieves certificate from CA.

Certificates in AG 2.0

- Windows
- Use the provided script to generate a request
 - %GLOBUS_LOCATION%\bin\certreq.cmd
- Start\Programs\Windows Globus\Get a Certificate
- Copy from a UNIX host using the Globus Configuration program

Globus Certificates

- Windows (con't)
- certreq.cmd
 - Use the defaults where provided
 - DNS domain (i.e., mcs.anl.gov)
 - Full Name (i.e., Ti Leggett)
 - Mail userreq.pem to the address reported back

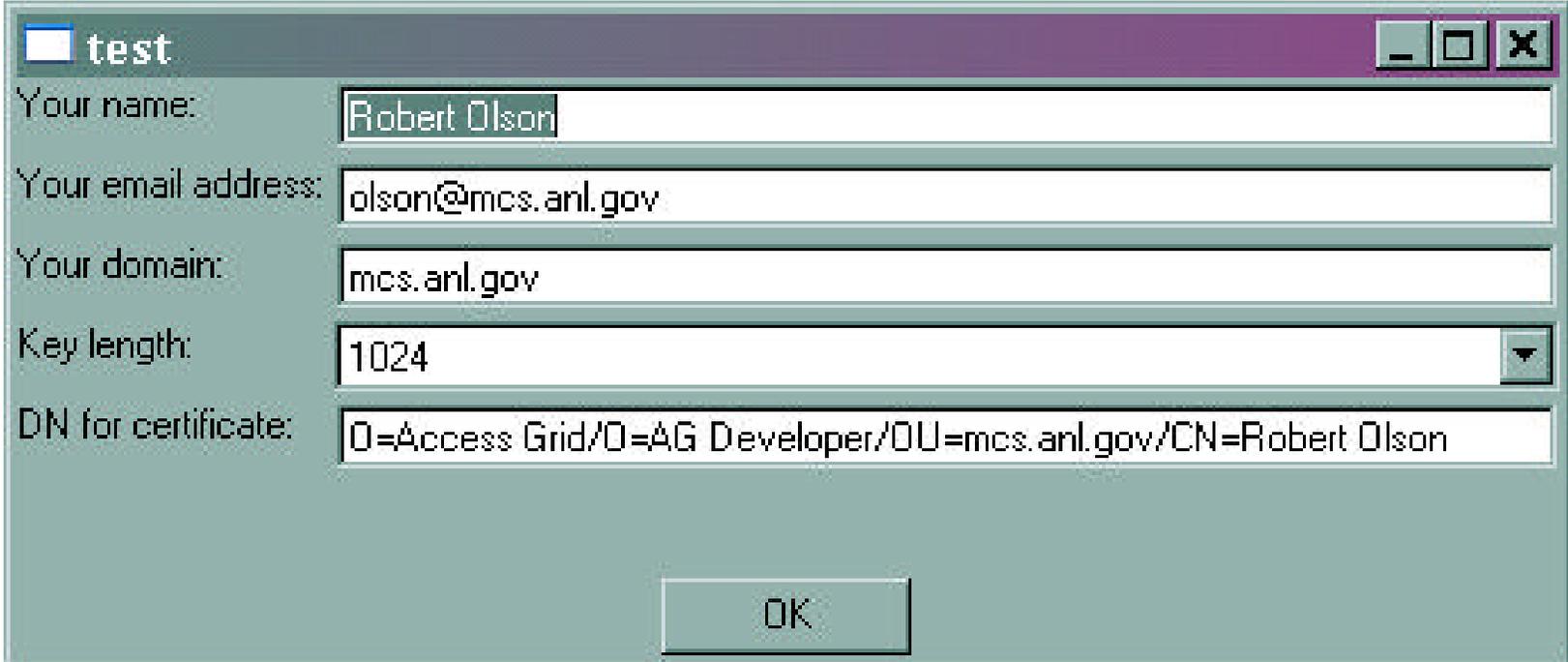
Globus Certificates

```
Get a Certificate
Using configuration from C:\Program Files\WinGlobus\bin\ssleay.conf
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....++++++
_++++++
writing new private key to 'userkey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name [Access Grid]:
Organizational Unit [agdev-ca.mcs.anl.gov]:
Enter your DNS domain []:mcs.anl.gov
Enter your full name, userid or other unique value which can
identify you within your organization
It may contain blanks []:Ti Leggett
ECHO is off.
-----
Your certificate request and key has been saved in
C:\Documents and Settings\leggett\Application Data\globus
Mail the userreq.pem to leggett@mcs.anl.gov
When the CA returns the certificate, save it as
C:\Documents and Settings\leggett\Application Data\globus\usercert.pem
-----
C:\Documents and Settings\leggett\Application Data\globus>_
```



Retreat 2003

New interface (snapshot)



A screenshot of a Windows-style dialog box titled "test". The dialog box has a purple title bar with standard minimize, maximize, and close buttons. It contains five input fields for certificate information and an "OK" button at the bottom center.

Your name:	Robert Olson
Your email address:	olson@mcs.anl.gov
Your domain:	mcs.anl.gov
Key length:	1024
DN for certificate:	O=Access Grid/O=AG Developer/OU=mcs.anl.gov/CN=Robert Olson



Retreat 2003

Installing your new certificate

- Cert will arrive in email
- Copy the body of the email to
 - C:\Documents and Settings\olson\Application Data\globus\usercert.pem

Key Management

- Private key critical to the authentication process
- If private key stolen, authentication of the certificate cannot be trusted
- Hence, private keys encrypted on disk
- Passphrase required to decrypt private key for each operation

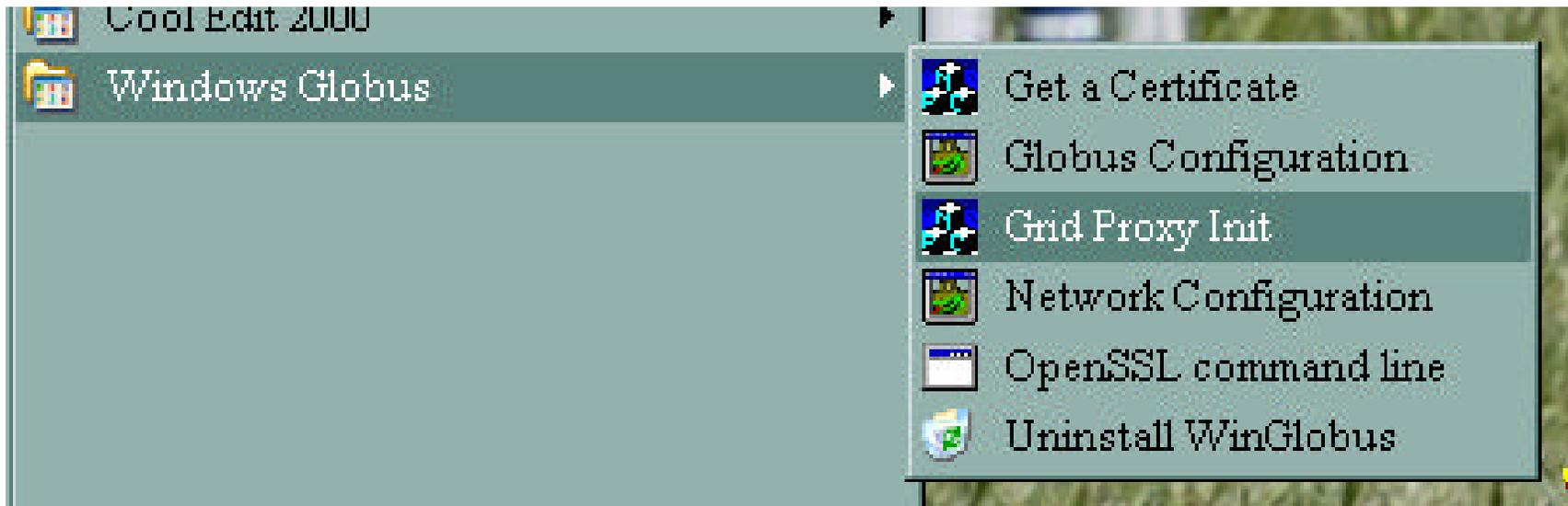
Single Sign-on

- Typing passphrase every time is not acceptable
- Globus uses *proxy certificates* to effect *single sign on*
- A proxy certificate is a specially formatted identity certificate with an unencrypted private key
- User's certificate acts like a miniature, special purpose CA to sign the proxy certificate
- Proxy can act on behalf of the user

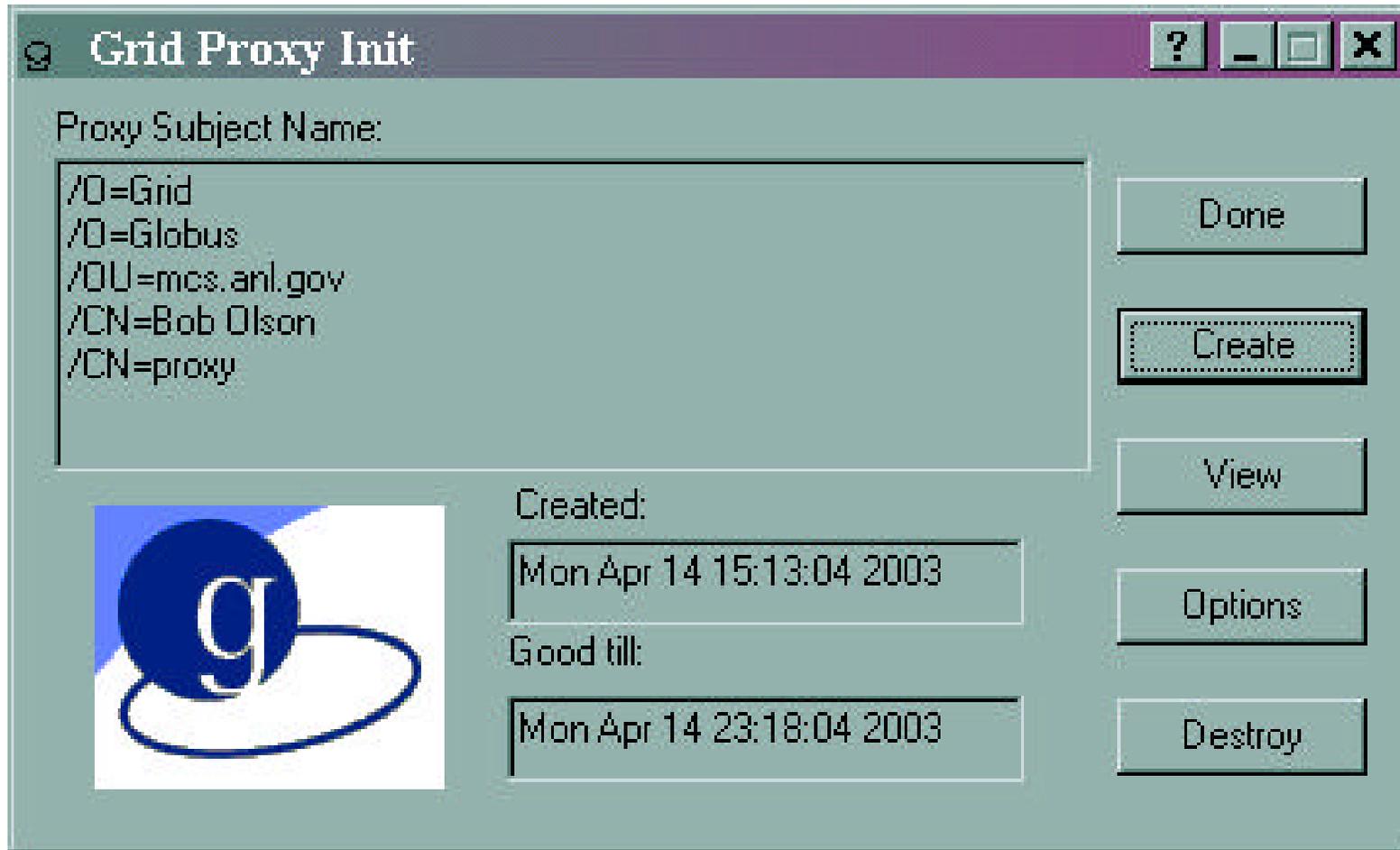
Single Sign-on, cont.

- How is this secure?
 - Lifetime of the proxy certificate limited (hours or small number of days)
 - Capability of the proxy can be limited
 - New features in late-model Globus
- Proxies not yet standard, but in process
- Special programs needed to create proxies
 - wgpi.exe
 - grid-proxy-init

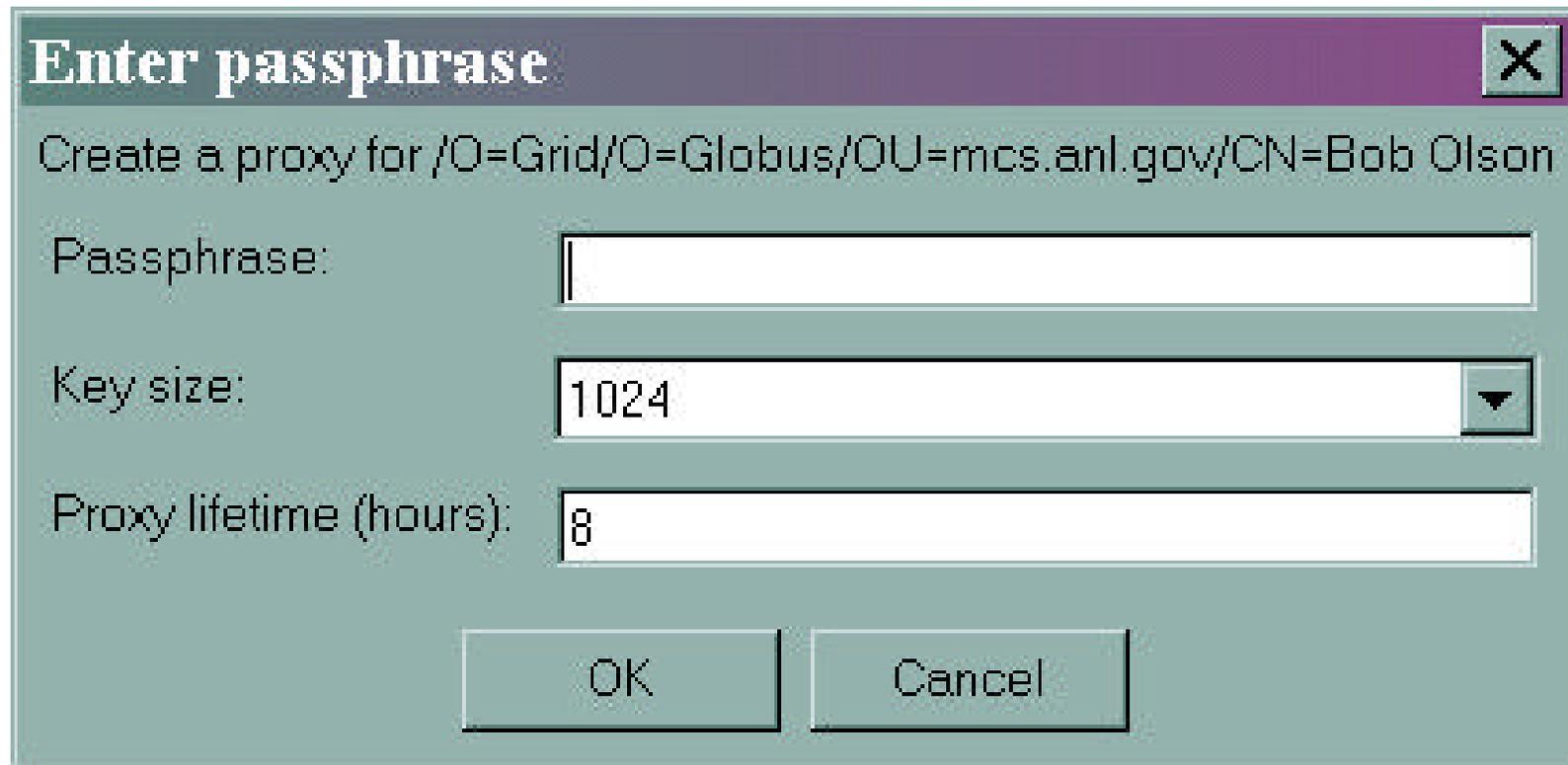
Creating proxy on Windows



Creating proxy on Windows, cont



New support for Certificate Management



Enter passphrase [X]

Create a proxy for /O=Grid/O=Globus/OU=mcs.anl.gov/CN=Bob Olson

Passphrase:

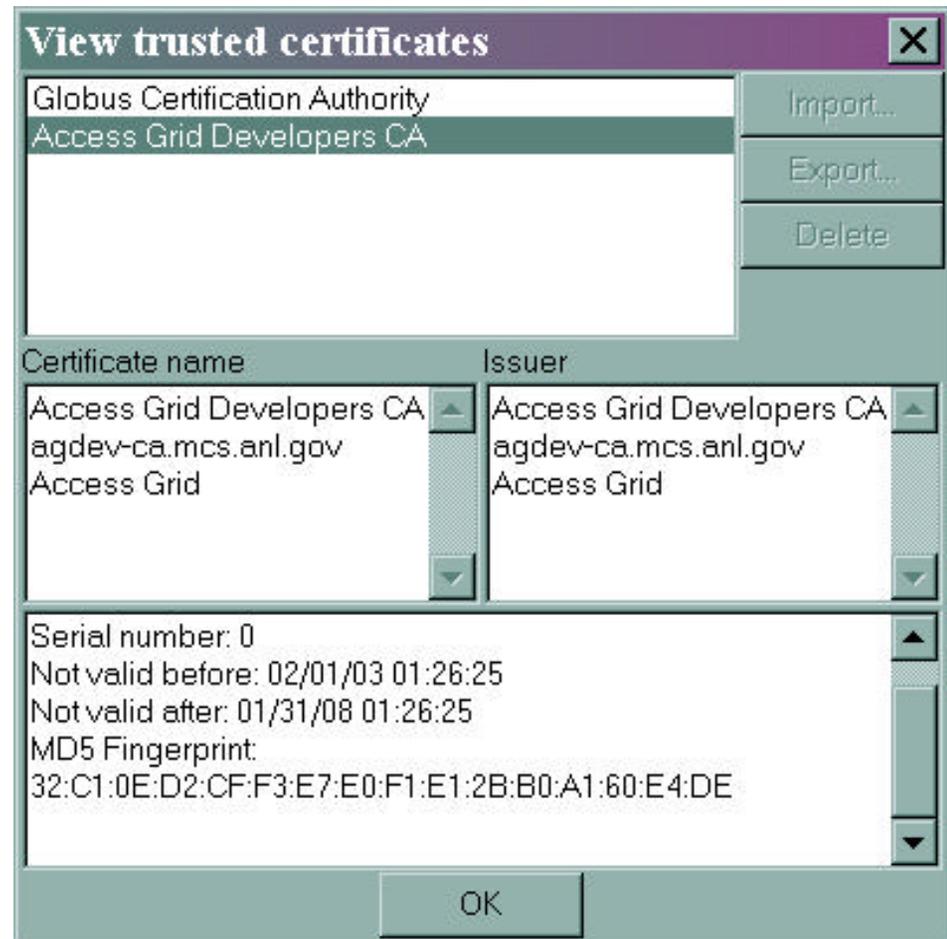
Key size: ▼

Proxy lifetime (hours):

OK Cancel

Soon, saner certificate management

- AG2.0 will manage certificates itself
- Current state allows browsing of certificates
- Import and export coming



Retreat 2003

OpenSSL

- The OpenSSL library provides tools for ...
 - Creating, querying X.509 certificates
 - Implementation of SSL transport protocol
- Usable from command line
- Usable as a C-based library
- Bound to Python via the pyOpenSSL library
- Used in Globus for SSL protocol support

OpenSSL, cont.

- We saw an example of an OpenSSL certificate dump earlier
- Create a keypair:
 - `openssl genrsa -out key.pem 1024`
- Creating a certificate request:
 - `openssl req -new -key key.pem -out req.pem`
- Extensive documentation on user apps and library
- www.openssl.org

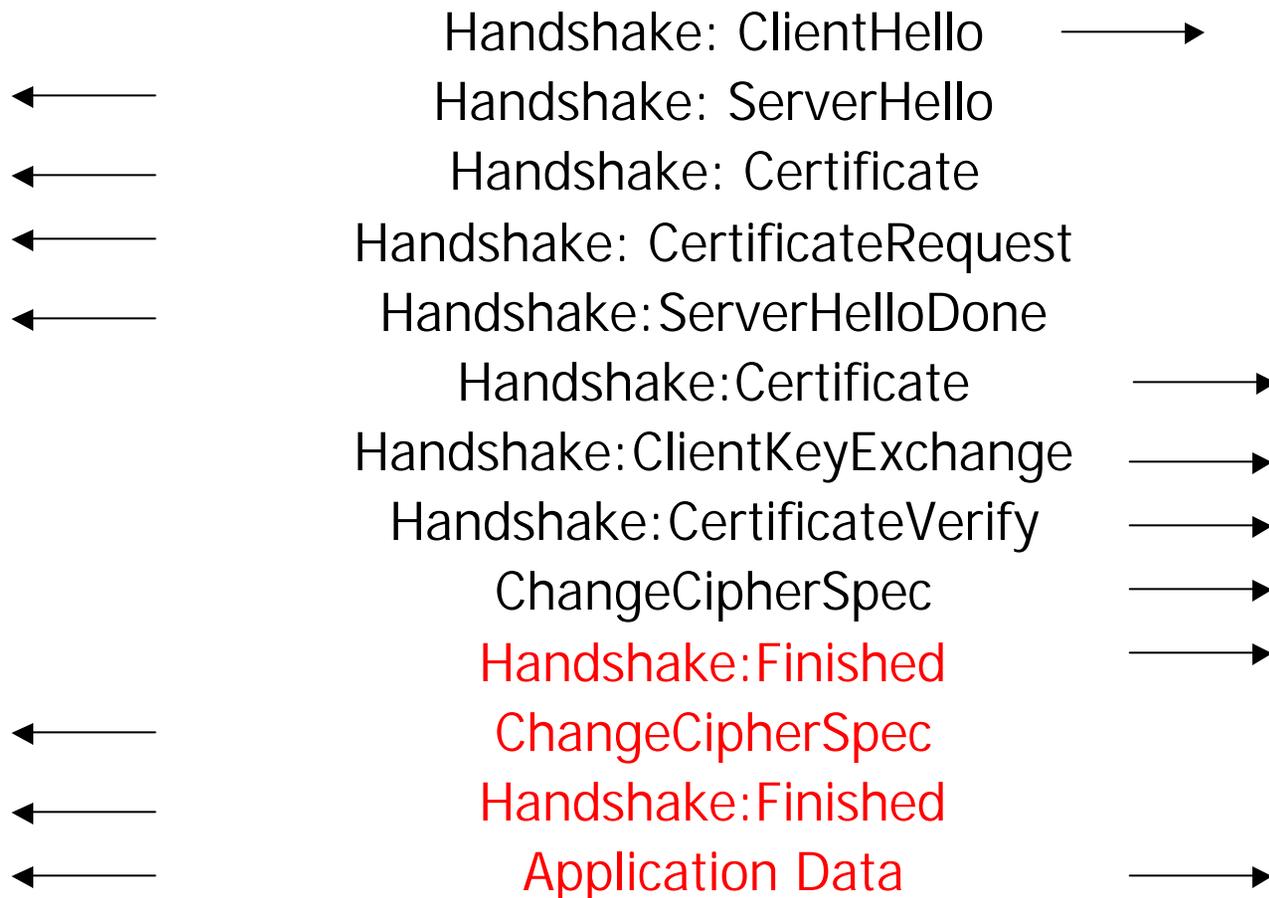
SSL

- Goal: Provide a point-to-point secure channel over insecure networks
- Uses public-key encryption to authenticate endpoints
- Endpoints negotiate common cipher and encryption key

SSL Handshake protocol

Client

Server



Retreat 2003

ssldump

- SSLv3/TLS network protocol analyzer
- Useful tool for debugging or learning about SSL connections in detail
- Built on libpcap (base for tcpdump)
- Sniffs traffic from the wire
- With available private keys, can display certificate info and encrypted traffic

Example trace from ssldump: connect to a venue

```
New TCP connection #7: moonbeam.mcs.anl.gov(56141) <->
vv2.mcs.anl.gov(9004)
7 1 0.0099 (0.0099) C>S Handshake
  ClientHello
    Version 3.0
    cipher suites
    SSL_RSA_WITH_NULL_MD5
    [...]
    compression methods
      NULL
7 2 0.0199 (0.0099) S>C Handshake
  ServerHello
    Version 3.0
    session_id[32]=
      63 3a e3 c1 1c a1 91 29 05 4b 84 d6 d0 44 86 d9
      bc e1 86 6e b7 ed 3a 15 c7 c0 45 a8 c1 f0 d6 66
    cipherSuite          SSL_RSA_WITH_NULL_MD5
    compressionMethod    NULL
```



Retreat 2003

Ssldump example, cont.

```
7 3 0.0199 (0.0000) S>C Handshake
Certificate
  Subject
    O=Grid
    O=Globus
    OU=mcs.anl.gov
    CN=Ivan R. Judson
    CN=proxy
  Issuer
    O=Grid
    O=Globus
    OU=mcs.anl.gov
    CN=Ivan R. Judson
  ...
  Extensions
    Extension: Netscape Cert Type
```



Retreat 2003

Ssldump example, cont.

Subject

C=US

O=Globus

CN=Globus Certification Authority

Issuer

C=US

O=Globus

CN=Globus Certification Authority

Serial 00

Extensions

Extension: X509v3 Basic Constraints

Extension: Netscape Cert Type



Retreat 2003

Ssldump example, cont.

```
7 4 0.0199 (0.0000) S>C Handshake
CertificateRequest
  certificate_types          rsa_sign
  certificate_types          dss_sign
  certificate_authority
    C=US
    O=Globus
    CN=Globus Certification Authority
  certificate_authority
    O=Access Grid
    OU=agdev-ca.mcs.anl.gov
    CN=Access Grid Developers CA
ServerHelloDone
```



Retreat 2003

Ssldump example, cont.

```
7 5 0.0599 (0.0399) C>S Handshake
Certificate
  Subject
    O=Grid
    O=Globus
    OU=mcs.anl.gov
    CN=Bob Olson
    CN=proxy
  Issuer
    O=Grid
    O=Globus
    OU=mcs.anl.gov
    CN=Bob Olson
  ...
  Extensions
    Extension: X509v3 Basic Constraints
    Extension: Netscape Cert Type
```



Retreat 2003

Ssldump example, cont.

```
7 6 0.0599 (0.0000) C>S Handshake
    ClientKeyExchange
7 7 0.0599 (0.0000) C>S Handshake
    CertificateVerify
    Signature[128]=
        6f 01 96 76 1e 14 07 6d 93 ad bb 06 12 4d b6 ce
        88 23 44 d5 6e cc 77 25 8e 47 33 e3 be 6c 19 bb
        71 25 75 12 3a 21 39 ca cb b4 f1 9e cf 27 e0 d0
        0f 7c 2c 1f b2 3b 15 38 bf 1d 1e a4 59 2f bb c0
        20 10 d1 63 00 84 f6 ab 95 21 e2 ea 7d 98 c3 e1
        bd 18 3b 78 ce 8f d2 3d 9e 94 89 39 d9 f2 51 5a
        c1 c9 0c b3 aa 4d 75 4b f5 5c 0b de 27 ee fe 2b
        8b 90 fd d9 37 71 4f 36 5d db 9e 62 af 51 f3 6f
7 8 0.0599 (0.0000) C>S ChangeCipherSpec
7 9 0.0599 (0.0000) C>S Handshake
7 10 0.1099 (0.0499) S>C ChangeCipherSpec
7 11 0.1099 (0.0000) S>C Handshake
7 12 0.1099 (0.0000) C>S application_data
```



Retreat 2003

Applying this to the Access Grid

- Discussion that follows is in the context of AG2.0 infrastructure
- AG1.0 had a very simple security architecture:
 - Username / password database on venue server
 - HTTPS via web browser and Apache for encrypted channel access
 - Distribution of shared session keys via HTTPS



Retreat 2003

Authentication

- Assumptions:
 - Authentication takes place on a transaction between a client and a server
 - Client and server each hold an identity cert
 - Authentication is *mutual*: After completion, client and server have verified identity of the other party
- Secured communications in AG2 use Globus...
- ...which uses SSL/TLS
- SSL/TLS defines protocol for a secure handshake with mutual authentication.



Retreat 2003

Authentication, cont.

- For any connection to succeed:
 - client must trust server's issuing CA
 - server must trust client's issuing CA
- Client software allows viewing of certificates of servers
 - Possibly problematic in current Globus toolkit
 - However, we can view the authenticated DN

Authorization

- Currently, venues maintain lists of DNs of users allowed entry
- In future, moving toward *role-based authorization* mechanisms.
- Users dynamically assigned to roles
- Access to resources given to roles

Confidentiality and Integrity

- Provided by underlying SSL transport
- All control traffic is private
- Media traffic secured by shared-key encryption (AES 128-bit currently)
- Shared keys generated by Venue and distributed over Globus-secured channels

Practical security issues

- In AG2.0, each user must have an identity certificate
- Identity certs issued by Certificate Authorities
 - AG Development CA
 - Globus test CA
 - DOE Science Grid CA
 - Commercial CA (Verisign, Thawte, ...)



Retreat 2003

Practical Security Issues, cont.

- Certificate Safety
 - If the private key for a cert is compromised, the cert cannot be trusted
 - Hence, users have responsibility for maintaining safety of their keys
- The use of identity certificates is often cumbersome

Identity Maintenance Alternatives

- NCSA MyProxy
 - Online proxy storage for standard identity certificates
 - Medium-term expiration proxies kept at central server
 - Proxies created via username/password authentication

Identity Maintenance Alternatives, cont.

- Online CA with username/password support
 - Identity certificates held at an online CA
 - Proxies created via username/password authentication
 - No requirement for user storage of certs
 - Integration with Shibboleth or other single sign-on infrastructure

Trust issues

- If a CA is not trusted by a service, then no certificates issued by that CA are trusted
 - CA trust is a *minimum requirement* for access
- Trusted CA certs must be configured on both clients and servers

Certificate Authorities

- Futures Lab dev group runs a CA for AG development
- /O=Access Grid
/OU=agdev-ca.mcs.anl.gov
/CN=Access Grid Developers CA
- Basic policy: requests arrive via email; if they look reasonable, certificate is issued
- Trust such certificates accordingly



Retreat 2003

Running your own CA

- Entirely possible
 - Raw OpenSSL-based CA
 - Good for low volume
 - Globus SimpleCA
 - Being phased out
 - OpenCA
 - Open source project under active developmt
 - Web-based interface
 - AG Development CA moving to OpenCA



Retreat 2003

Using another CA

- Various large collaborations run CAs
 - DOE Science Grid
 - NCSA Alliance
- Commercial CAs
 - Verisign
 - Thawte
- To trust, need the CA's public key certificate



Retreat 2003

Configuring Trusted CAs

- Globus defines a directory containing trusted CA certs
- Linux
 - `/etc/grid-security/certificates`
- Windows
 - `\Program Files\Winglobus\Certificates`



Retreat 2003

Trusted CAs, cont.

- Each trusted CA represented by two files:
 - CA Certificate
 - CA signing policy file (defines to the Globus toolkit the namespace of certificates issued by this CA)
- CA cert files named by the hash of the subject DN, followed by a index
- Signing policy named <hash>.signing_policy
- (What if two certs hash to the same string?)

Example

```
$ cd /etc/grid-security/certificates/  
$ ls  
42864e48.0  
42864e48.signing_policy  
45cc9e80.0  
45cc9e80.signing_policy  
$ openssl x509 -issuer -subject -hash -in  
  42864e48.0 -noout  
issuer= /C=US/O=Globus/CN=Globus  
  Certification Authority  
subject= /C=US/O=Globus/CN=Globus  
  Certification Authority  
42864e48
```



Retreat 2003

Adding a new Trusted CA

- Globus SimpleCA
 - For Linux clients, SimpleCA provides an installation package to update trusted CA list
 - For Windows, manually copy CA cert and signing policy file to \Program Files\WinGlobus\certificates

Certificate Revocation

- In a PKI, need to support the revocation of invalid certificates
 - Private key compromised
 - User no longer at institution
 - CA key compromised
- Globus supports *Certificate Revocation List* (CRL).
- Similar to the old bad credit card number books, with similar problems

This is all very complex!

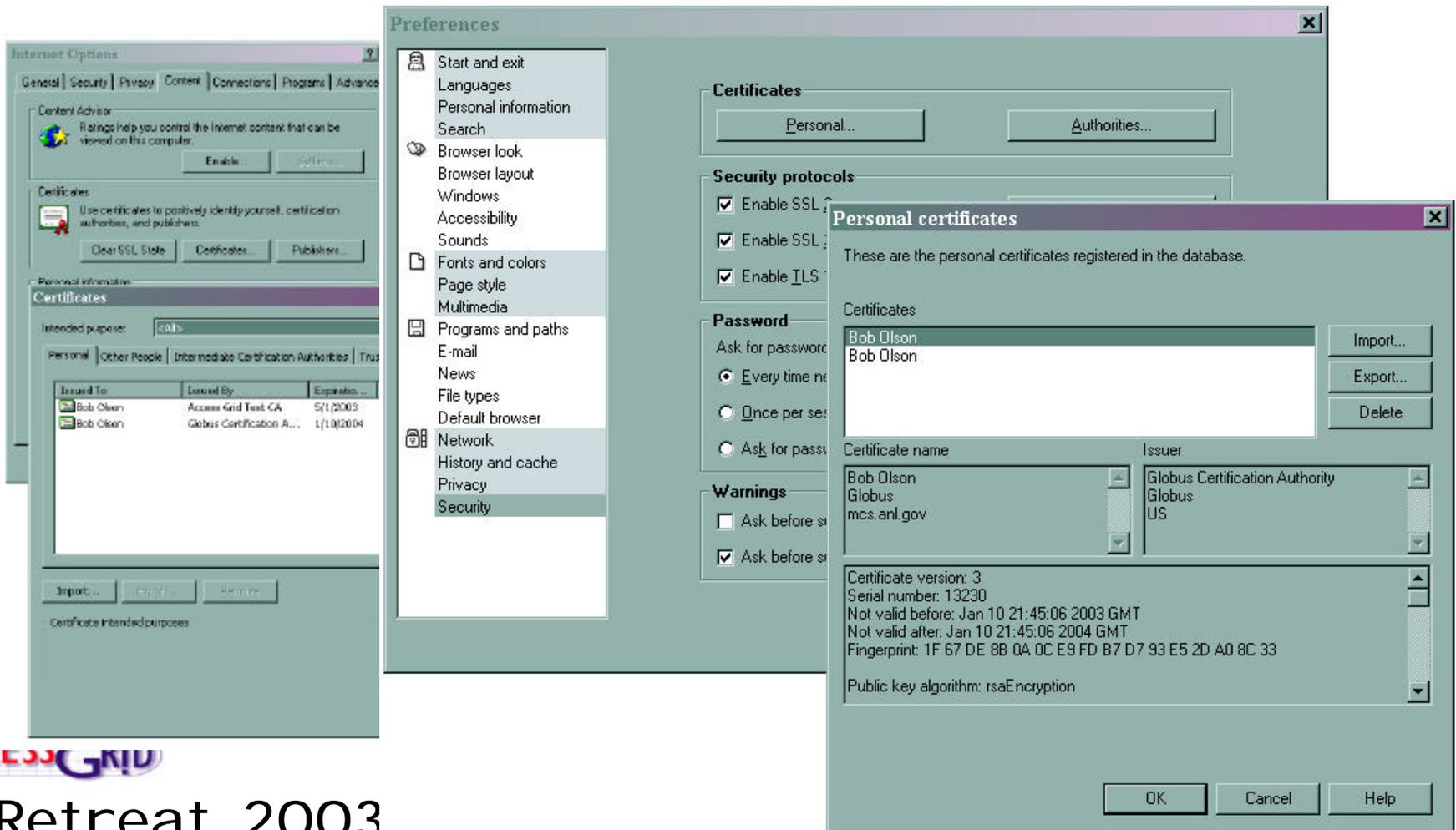
- Certificates have drawn the most questions so far in the AG2 project
- How solved in other projects?
 - Globus assumed Unix environment, tightly-controlled CA model, same identity everywhere (Users had to learn lots of details)
 - Web browsers provide server-side authentication, people never think about the actual server certs
 - Did you know there is a compiled-in set of trusted CA certs in your browser?
 - Client-side authentication available but almost never used in web browsers



Retreat 2003

Complex, cont.

- However, browsers provide useful model for dealing with certificates



Complex, cont.

- Current plan: Emulate web browsers
- Provide interfaces for manipulating certificates
 - Browse
 - Import
 - Export
- Automate the installation of trusted CA certs
- Integrate the certificate request, distribution, and installation processes

Development and Security

- For the most part, security provided transparently by AG toolkit library
- Detailed certificate manipulations provided by CertificateManager object
 - Management of *certificate repositories*
 - Management of per-application configuration state (default identity, etc)
 - Configuration of runtime environment for Globus use
 - Creation of Globus proxies
 - Integration with GUI and non-GUI applications



Retreat 2003

Certificate Repository

- Application-maintained set of certificates
- AG Client currently uses two:
 - Identity certificate repository
 - Trusted CA certificate repository
- Provides interfaces for GUI browsing of certificates
- Provides certificate lookup
 - (Runs into DN uniqueness issues; not completely resolved yet)



Retreat 2003

Certificate Repository, cont.

- Planned:
 - Support for multiple identity certificates.
 - Support for multiple active proxy certificates
 - Automated (or assisted) choosing of certificate based on target venue
 - Support for MyProxy

Certificate wrapper

- Certificate class provides a wrapper for underlying OpenSSL interface code
- High-level interface for basic certificate operations
 - Get subject, issuer
 - Validity tests

Runtime environment issues

- External apps needs find identity certificates, Globus proxy, trusted CA certs
- Python code can use CertificateManager to handle this
- Non-python code can either handle it manually (certificate architecture document, in the release, discusses this)
- Or it can be executed from a Python app that uses the CertificateManager to bootstrap state

Runtime issues, cont.

- Apps that are launched from the Venue client will have security environment preconfigured
- Open issues remain here: interested in developer feedback

Runtime issues, cont.

- Beware third-party code!
- None allowed in venue server
- Be very wary of allowing in venue client
- Isolation of third-party code in separate processes, use of proxy certs valuable
- Think Trojan horse, virus, DDOS, backdoor, etc etc etc.
- Consider site security policies

Final Comments

Cryptography by itself is fairly useless.

Ferguson and Schneier

- Crypto tools and libraries are not a panacea
- Nearly all successful attacks are not against the crypto itself
- (Find the serious problem in the current AG2 beta code and I'll buy you a {Coke,beer})



Retreat 2003

Recommended Reading

- Ferguson, N. and B. Schneier, *Practical Cryptography*. 2003: Wiley Publishing, Inc.
- Schneier, B., *Applied Cryptography*. 1996: John Wiley and Sons, Inc.
- Anderson, R.J., *Security Engineering: A Guide to Building Dependable Distributed Systems*. 2001: John Wiley and Sons, Inc.
- Schneier, B., *Secrets and Lies, Digital Security in a Networked World*. 2000: John Wiley and Sons, Inc.
- CryptoGram newsletter,
<http://www.counterpane.com/crypto-gram.html>



Retreat 2003