UNIVERSITY of STIRLING

www.cs.stir.ac.uk

Web Services

RESTful Services and Security

1

RESTful Services

- All the web services considered so far:
  - Use SOAP for communication with the service
  - Have service interfaces defined in WSDL
- Some developers believe that the overheads and complexity of this kind of web service are undesirable
- REST (Representational State Transfer) has been introduced more closely aligned with the original design of the web

2

RESTful Services

- Web contains resources
- Resource is an item of interest, e.g. html pages, images
- E.g. Amazon online store holds a resource for a new digital camera. Clients can access this resource at www.amazon.co.uk/canon/dslr
- On clicking the link, a representation of the resource is returned to the client (canon_dslr.html). The representation places the client into a certain state
- The result of the client traversing a hyperlink in canon_dslr.html is it access a second resource which places the client into a different state
- Thus the client application transfers (changes) states with each resource representation downloaded
- Consequently, the web is a REST system
- You have been using REST services for some time!

RESTful Services

- Web servers support four basic methods collectively known as CRUD:
  - Create, supported by POST in HTTP
  - Read, supported by GET in HTTP
  - Update, supported by PUT in HTTP
  - Delete, supported by DELETE in HTTP
- Of these, GET (used in ordinary web browsing) and POST (often used with forms) are the best-known
- The four HTTP operations are sufficient to implement a wide variety of web services

RESTful Services

- A key idea in REST is that a URI should identify a resource on which operations can be performed, e.g.:
- An online shop has stocks considered as a resource; the whole stocks might be identified by the URI www.shop.com/stocks
- Each product is itself a resource within the overall stocks; one product might be identified by the URI www.shop.com/stocks/product4059
- HTTP methods can then be used to create, read, update or delete the whole stocks or an individual product in stock
- Like other web services, RESTful services normally send and receive XML documents (though this is not essential)
- RESTful services are best suited for database type systems
- Simple datatypes can be transmitted as part of the request URI (GET)
- Complex datatypes can be transmitted as message payload (XML), but cannot be accessed with the GET message

RESTful Services - Parameters

A service like this cannot be accessed with HTTP GET:

```
public class Man {
   String name;
   int age;
   Address address
}
public String getName(Man man) { //doSomething}
```

But a service like the following can easily be accessed using HTTP GET:

```
public String getName(String id, int age) {
//doSomething}
```

http://ws.stir.ac.uk:8080/axis2/services/NameService/getName?id=Paul&age=10

RESTful Services

- Opinions are split on REST vs. SOAP; in fact it is possible to create web services that support both:
  - Major supporters of REST include del.icio.us, Flickr and Yahoo
  - Major supporters of SOAP include Google
  - Major companies supporting both include Amazon and eBay
- Several toolsets support REST, e.g.:
- Axis2 is a widely used tool for supporting both REST and SOAP messaging
- JDK (Java Development Kit, java.sun.com/javase) has packages (e.g. javax.xml, javax.xml.ws) that support XML in general and REST in particular
- JAX-WS (Java API for XML – Web Services, jax-ws.dev.java.net) defines an API for Java programs to use REST, also offers a REST-compatible application server called GlassFish
- Restlet (www.restlet.org) provides Java support for developing RESTful services

Requirements for web service security

- Security requirements are diverse, depending on the purpose of the web service application
  - Maybe not required at all
  - Full security for commercial operations
- Needs to be open and extensible
- Needs to be interoperable and work between different organisations
- Should leverage existing standards

Security for WS

- A family of related standards has been developed to address three major issues
- **WS-Security** (**Web Services Security**)
  - provides secure transfer over SOAP
- **WS-Trust** (**Web Services Trust**)
  - allows organisations to validate each other, and to develop chains of trust
- **WS-SecureConversation** (**Web Services Secure Conversation**)
  - uses security tokens to allow efficient encryption of messages

UNIVERSITY of STIRLING

Slide 9

9

Security for WS

- Encryption for secure transfer of data must meet the following challenges:
  - eavesdropping on, interference with, and reply of messages
  - authentication of the communicating parties
  - non-repudiation (no denial) of messages having been sent

UNIVERSITY of STIRLING

Slide 10

10

Security for WS

- A widely used approach is based on PKI (Public Key Infrastructure)
- Encryption and decryption use inverse algorithms that make encryption with the public key easy, but decryption without the private key very difficult
  - A receiver publishes a public key for messages it should receive
  - A sender uses this key to encrypt messages
  - A receiver also has a private key that it uses to efficiently decrypt messages
  - Although the public and private keys are related, it should be impracticable to determine the private key from the public key

UNIVERSITY of STIRLING

Slide 11

11

Security for WS

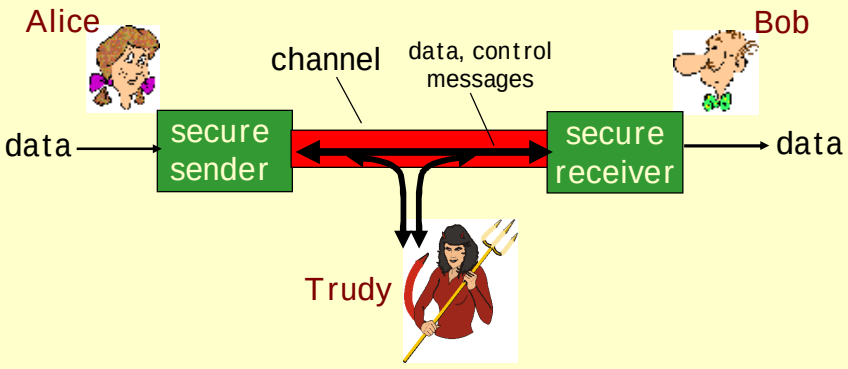
- Digital certificates are widely used to establish the authenticity of a party:
  - CA (Certificate Authority) issues parties with digital certificates that guarantee they are who they claim to be
  - For large-scale use, certificates are signed by chains of CAs starting from a root CA that everyone trusts (e.g. Thawte, Verisign)
  - Chain of trust might be Verisign, JISC (Joint Information Services Committee), University of Stirling, M. Kolberg
  - HTTPS, HTTP over SSL (Secure Sockets Layer), employs digital certificates for secure use of the web

UNIVERSITY of STIRLING

Slide 12

12

Case study - introduction

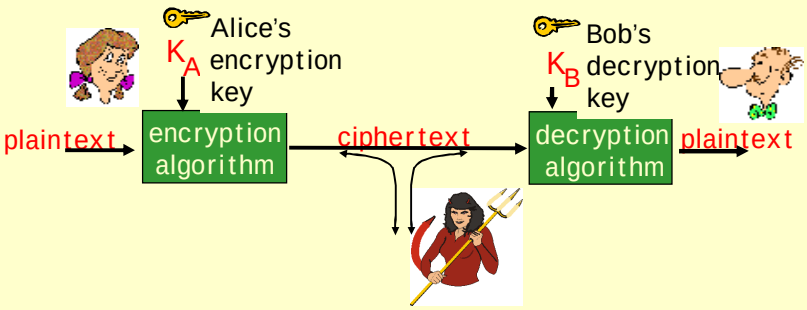


UNIVERSITY of STIRLING

Slide 13

13

General Cryptography



symmetric key crypto: sender, receiver keys *identical*

public-key crypto: encryption key *public*, decryption key *secret* (private)

UNIVERSITY of STIRLING

Slide 14

14

Symmetric key cryptography

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another
- E.g. add 3 to ASCII to encrypt, subtract 3 to decrypt
- Encrypter and decrypter are the same (i.e. not "one-way")
- Fast! – good for bulk en/decryption
- More complex algorithms & larger keys slow the process down, but increase security

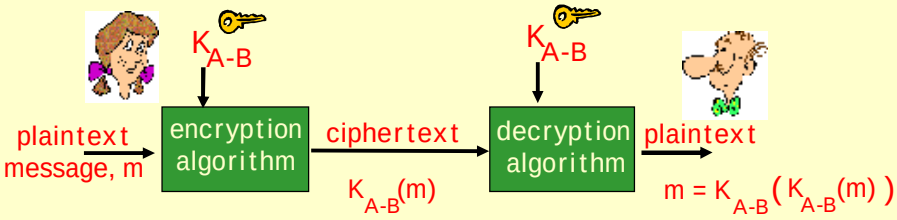
```
plaintext:  abcdefghijklmnopqrstuvwxyz
ciphertext: mnbvcxzasdfghjklpoiuytrewq

e.g.:  Plaintext: bob. i love you. Alice
      ciphertext: nkn. s gktc wky. mgsbc
```

UNIVERSITY of STIRLING

Slide 15

15

Symmetric key cryptography



Bob and Alice share know same (symmetric) key: K

- E.g., key is knowing substitution pattern in mono alphabetic substitution cipher
- How do Bob and Alice agree on key value?

UNIVERSITY of STIRLING

Slide 16

16

Public Key cryptography

symmetric key crypto

- Requires sender, receiver know shared secret key
- How to agree on key in first place (particularly if never "met")?

public key cryptography

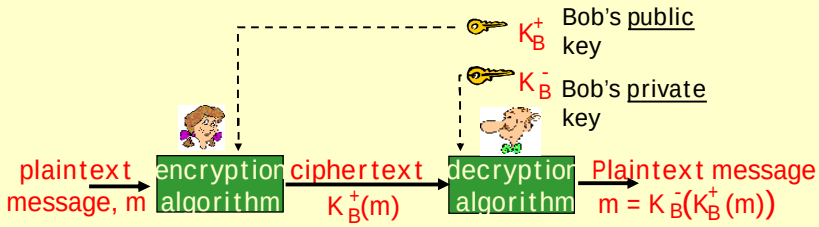
- radically different approach
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

UNIVERSITY of STIRLING

Slide 17

17

Public key cryptography



- Keys A and B are different & related → cannot be derived from the other
- Either key may be used for encryption, only the other key can decrypt the message (encryption is one-way!)

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

- If a message is decryptable by one key, it could only be have been encrypted using the other key – GUARANTEED!

UNIVERSITY of STIRLING

Slide 18

18

Public key cryptography

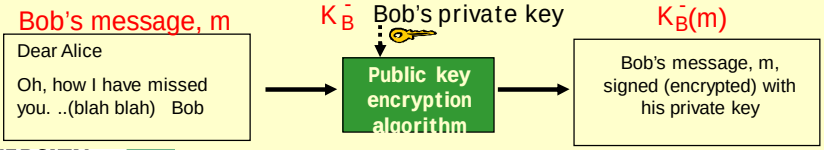
- Very large key sizes are used
  - JSSE has RSA (Rivest, Shamir, Adelson algorithm) key scheme with 2048 bit keys
- Disadvantages: encryption and decryption are slow
  - Not useful for bulk data
  - But good for authentication and key agreement
- Usage:
  - Obtain a unique pair of keys
  - Keep private key private, make public key available on "who needs to know" basis

Confidentiality & Authentication

- A sends a message to B encrypted using B's public key
  - Only B can decrypt using private key (confidentiality)
- B sends a message to A encrypted with B's private key
  - Anyone who knows B public key can decrypt it (not confidential, but useful for authentication)

Message integrity – Digital Signatures

- Cryptographic technique analogous to hand-written signatures.
- Small piece of information added to a message
  - Bob digitally signs document (private key), establishing he is document owner/creator.
  - **verifiable, nonforgeable**: Alice can prove that Bob, and no one else, must have signed document (decrypting message using Bob's public key)
  - Document was not damaged in transit (Bob signed m and not m')
- Attacker can decrypt signature, but not create new, correct one



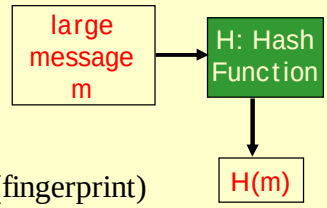
UNIVERSITY of STIRLING

Slide 21

21

Message Digests

- Computationally expensive to encrypt long messages
- Goal: fixed-length, easy- to-compute digital "fingerprint"
- Apply hash function H to *m*, get fixed size message digest, *H(m)*.



Hash function properties:

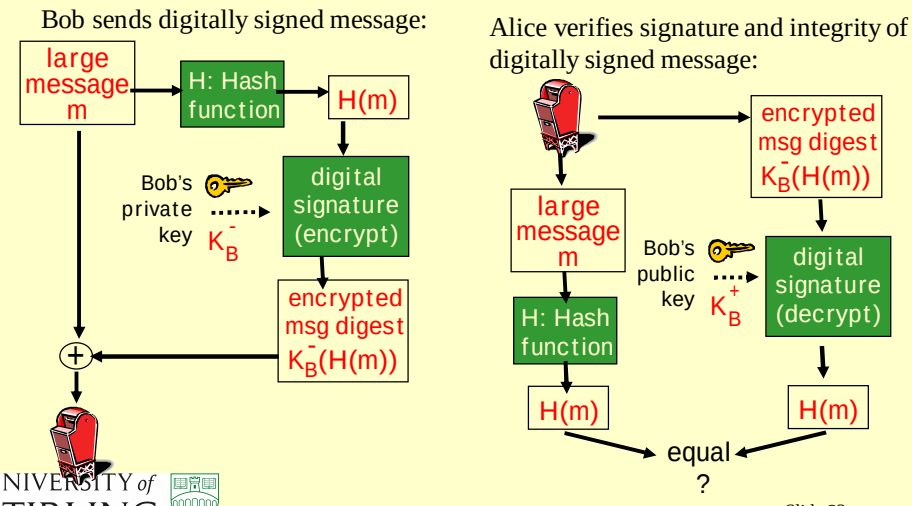
- many-to-1
- produces fixed-size message digest (fingerprint)
- given message hash x, computationally infeasible to find m such that x = H(m)

UNIVERSITY of STIRLING

Slide 22

22

Signed message digest



UNIVERSITY of STIRLING

Slide 23

23

Public Key Problem

- When Alice obtains Bob's public key (from web site, e-mail, diskette), how does she know it is Bob's public key, not Trudy's?

Solution:

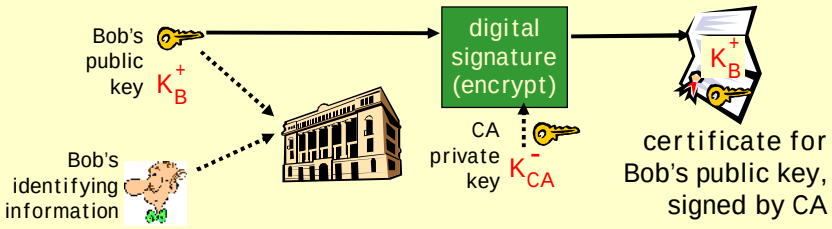
- trusted certification authority (CA) which binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E's public key digitally signed by CA – CA says "this is E's public key"

UNIVERSITY of STIRLING

Slide 24

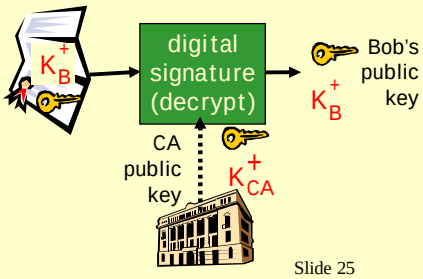
24

Digital Certificates



When Alice wants Bob's public key:

- gets Bob's certificate (Bob or elsewhere).
- apply CA's public key to Bob's certificate, get Bob's public key



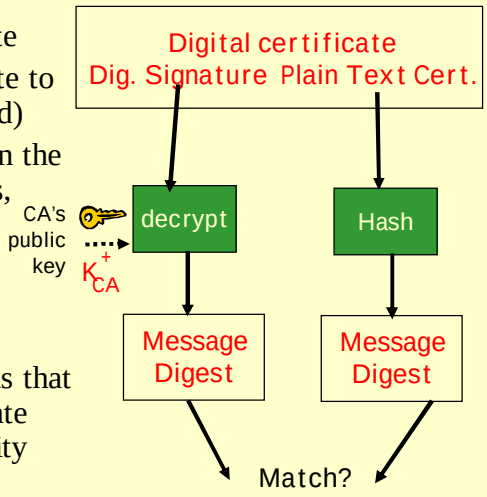
UNIVERSITY of STIRLING

Slide 25

25

Authentication …

- Server holds digital certificate
- Server sends digital certificate to client (no encryption required)
- Client looks up CA's name in the digital certificate and obtains, CA's public key
- Client verifies the digital signature attached to the certificate
- Successful verification means that the public key in the certificate really does belong to the entity named in the certificate



UNIVERSITY of STIRLING

Slide 26

26

The complete picture …

- Finally we have a complete apparatus:
  - Connection
  - Server sends digital certificate
  - Client verifies digital certificate & extracts server's public key
  - Client chooses a new key for symmetric encryption
  - Client encrypts the new key with server's public key & sends it to server
  - Client and server both know the symmetric key and encrypt subsequent communications using it
- This is how Secure Socket Layer (SSL) works as the base for PKI

UNIVERSITY of STIRLING

Slide 27

27