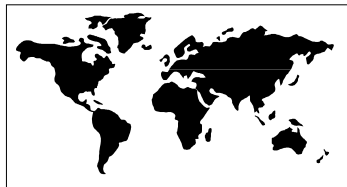


WWW - Grundlagen und Technologie

Sicherheitsaspekte, Abschluss und Ausblick



Erik Wilde
TIK - ETH Zürich
Sommersemester 2001

Übersicht

- Cookies (HTTP State Management)
 - was man damit machen kann und was nicht
- Grundbegriffe der Sicherheit
 - grundlegende Algorithmen
 - Anwendungen in gängigen Technologien
 - Zertifikate und *Public Key Infrastructures (PKI)*
- Zahlungssysteme
 - Credit Card Payments (SSL und SET)
 - digitales Geld
- Diplom- und Semesterarbeitsthemen
- Zusammenfassung

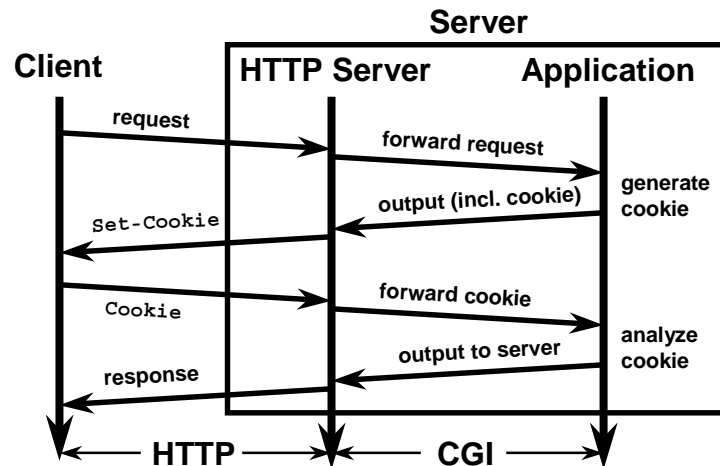
Cookies

- Anwendungen brauchen oft Benutzerstatus
- Realisierung durch Information, die vom Server zum Browser und zurück geht
 - Kompensation des *stateless approach* von HTTP
 - ermöglicht Server, Statusinformationen für Benutzer zu speichern
 - Speicherung eines Tokens beim Benutzer
 - daher immer abhängig von einer Browser-Installation
- oft verwendet, selten zu merken
 - *shopping cart* für on-line Bestellsysteme
 - Personalisierung von Seiten
 - *Website tracking* für persönliche Benutzerprofile

Cookies und HTTP

- RFC 2169 ("HTTP State Management")
- *Set-Cookie* Response Header Field
 - setzt (speichert) einen Cookie beim Client
 - kann vom Client ignoriert werden
- *Cookie* Request Header Field
 - Client sendet Cookie zum Server
 - Berücksichtigung der URL
 - Berücksichtigung des Alters des Cookies
- Cookies werden nicht ausgeführt
 - keine destruktive Gefahr wie Viren
 - jedoch Preisgabe sensibler Daten

Cookie Handling



WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

5

Security needs on networks

- **Confidentiality:** Only authorized people (eg, the sender and recipient of a message, and not any eavesdroppers) can know the message
- **Authentication:** When Bob receives a message that purports to be sent by Alice, Bob can be sure that the message was really sent by Alice
- **Integrity:** When Bob receives a message, he can be sure that it was not modified en route after Alice sent it.
- **Non-repudiation:** Alice cannot later deny that the message was sent. Bob cannot later deny that the message was received

WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

6

Grundlegende Algorithmen

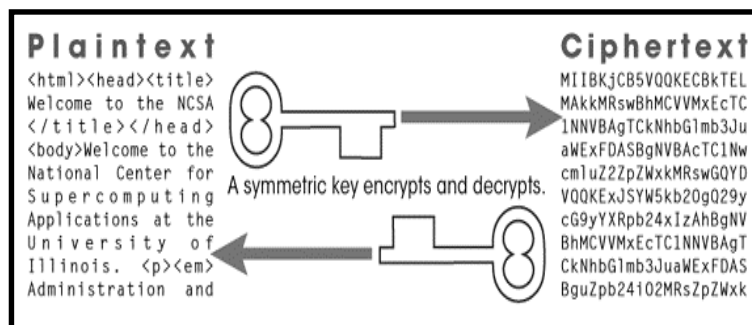
- Symmetrische Algorithmen
 - älteste bekannte Algorithmen
 - einfachstes Beispiel ist der Cäsar-Code
- Asymmetrische Algorithmen
 - wesentlich moderner (seit 1970)
 - geeignet für neue Anwendungsfelder
- Einweg-Funktionen
 - Verwendung für Diffie-Hellman Key Agreement
 - Verwendung für digitale Signaturen
 - essentiell für kryptographische Sicherheit

WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

7

Symmetrische Algorithmen

- ebenso bekannt als *Secret Key Verfahren*
- verwenden einen Schlüssel
 - Ver- und Entschlüsselung mit gleichem Schlüssel
 - altbekanntes Problem des Schlüsselaustauschs



WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

8

Data Encryption Standard (DES)

- designed by IBM in 1975, with help from NSA
- 56 bit keys, so there are 2^{56} keys, or about 70,000,000,000,000,000
- 2^{56} is a big number, but not that big. In August 1998, the Electronic Frontier Foundation demonstrated that a special-purpose machine built from standard parts at a cost of \$200,000 could break DES in 56 hours
- big governments have a lot more than \$200,000 to spend on cryptanalysis

Secure key distribution is critical

- with a symmetric system like DES, Alice and Bob have to agree on a shared secret
 - doesn't work (well) on a large scale
 - doesn't work with people who haven't met in advance
- Diffie-Hellman key agreement (1976)
 - Alice and Bob can create a shared secret key by exchanging messages, even though everyone can eavesdrop on the messages!

Lange Zeit ungelöstes Problem

- wie kann man sicher kommunizieren
 - ohne gemeinsamen Geheimnis
 - ohne sicheren Kommunikationskanal
- Schlüsselverteilung vorher immer *out-of-band*
- Grundidee von Diffie-Hellman
 - Alice verschliesst eine Kiste an Bob
 - Bob macht das eigene Schloss hinzu
 - Alice entfernt ihr Schloss
 - ...und schon ist das Problem gelöst!
 - ...aber leider sind mathematische Operationen oft nicht so einfach austauschbar in ihrer Anwendung...

Diffie-Hellman Grundidee

- Find a one-way function, that is, a function that is quick to compute, but slow to invert
 - Example: Multiplication and factoring - You can multiply two numbers in time proportional to the number of digits. But (as far as anyone knows), the time required to factor a number grows as the size of the number. So, we could quickly multiply a pair of 500 digit numbers. But if we give people the product, it will take them on the order of 10^{500} times as long to factor the number as it took us to do the multiplication

Diffie-Hellman One-Way-Function

- Modular exponentiation: Given a prime p , and numbers a and w less than p , compute $y = a^w \bmod p$ (can be done in $\log_2 w$ steps.)
- Discrete log problem: Given p , a , and y , find a w such that $y = a^w \bmod p$ (requires time on the order of p , as far as anyone knows.)
- So if we take p to be a 500 digit prime, the difference between the computing effort to compute powers mod p versus computing discrete logs mod p is on the order of 2^{500}

Diffie-Hellman

Alice:

- wähle Werte für p and a
 - öffentlicher Austausch
- wähle geheimes w
 - berechne $y = a^w \bmod p$
- übertrage y an Bob
- berechne $z^w \bmod p$

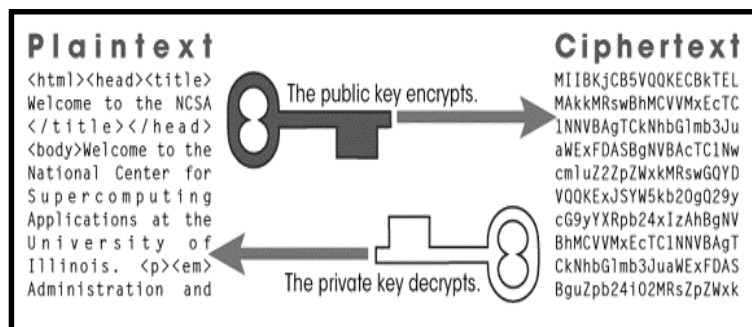
Bob:

- wähle Werte für p and a
 - öffentlicher Austausch
- wähle geheimes v
 - berechne $z = a^v \bmod p$
- übertrage z an Alice
- berechne $y^v \bmod p$

- das Resultat der Berechnungen ist das gleiche!
- der Grund: $(z^w = (a^v)^w = a^{vw} = a^{wv} = (a^w)^v = y^v) \bmod p$

Asymmetrische Algorithmen

- ebenso bekannt als *Public Key* Verfahren
- verwenden prinzipiell Schlüsselpaare
 - private key ist nur dem Eigentümer bekannt
 - public key kann allgemein verteilt werden



WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

15

Digitale Signaturen

- Verwendung von *One-Way Hash Functions*
 - oftmals auch als *message digest* bezeichnet
 - Erzeugung eines *digitalen Fingerabdrucks*
 - Umkehrrichtung nicht möglich
- Verschlüsselung des Fingerabdrucks
 - Verwendung des *private key*
- Senden von Dokument und Resultat
- Empfänger überprüft Echtheit des Dokuments
 - selbständige Berechnung des Fingerabdrucks
 - Entschlüsselung des empfangenen Fingerabdrucks mit Hilfe des *public key*
 - Vergleich beider Werte

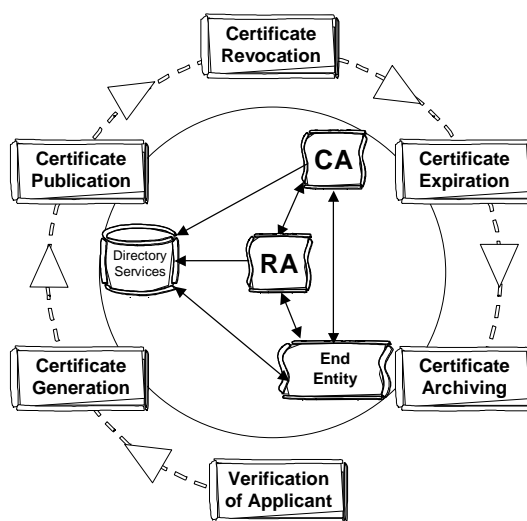
WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

16

Zertifikate

- bestätigen die Echtheit eines Schlüssels
 - öffentlicher Schlüssel des Zertifikatsbesitzers
 - Identifikationsinformation (z.B. Personenangaben)
 - digitale Signatur der *Certification Authority (CA)*
- entsprechen offiziellen Dokumenten
 - Pass zertifiziert die Identität einer Person
- in der Schweiz erste CA die Swisskey AG
- Problem: wer dient als CA für CAs?
 - keine einfache Lösung möglich
 - z.B. Zertifikat von Swisskey akzeptieren

Prozesse im PKI Umfeld



CA - Certification Authority

- Controls policy
- Generates certificate
- Manages revocation lists
- Updates directory
- Protects issuer (CA) keys

RA - Registration Authority

- Identifies end entity
- Allocates roles
- Interface to end entities

DS - Directory Services

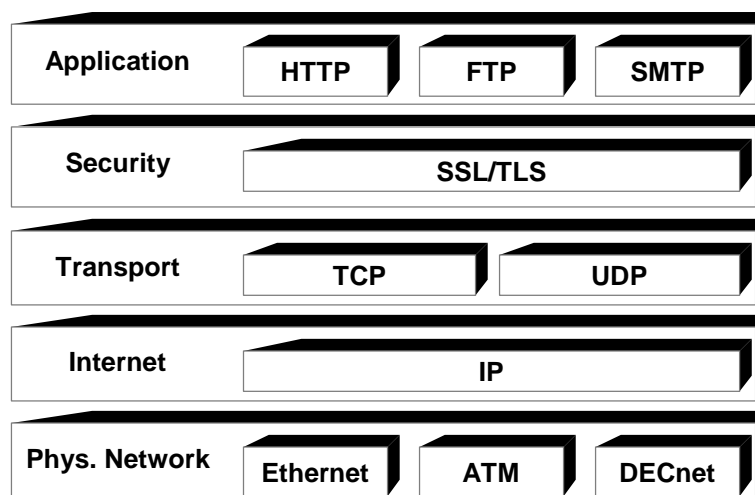
- Publishes end entity information

End Entities
(users, apps, etc.)

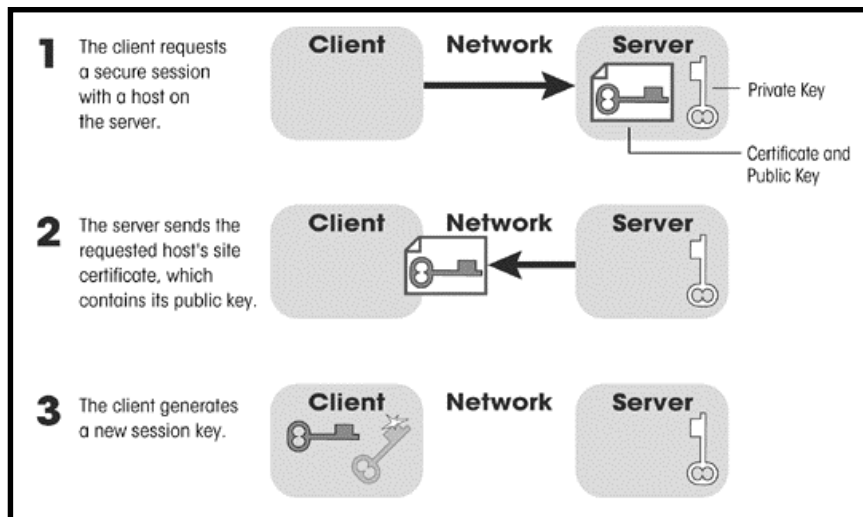
Secure Sockets Layer (SSL)

- z.Z. weitestverbreitete Sicherheitstechnologie
- Web-Server akzeptiert TCP/IP und TCP/IP über SSL Verbindungen
 - Kennzeichnung sicherer Dokumente mit dem `https:` URL Prefix
 - Standard-Verbindungen auf Port 80
 - SSL-Verbindungen auf Port 443
- aktuelle Version ist SSL3 (Verbesserungen gegenüber SSL2, mehr Algorithmen)
- Internet Standardisierung als *Transport Layer Security (TLS)*

Internet Protocol Suite



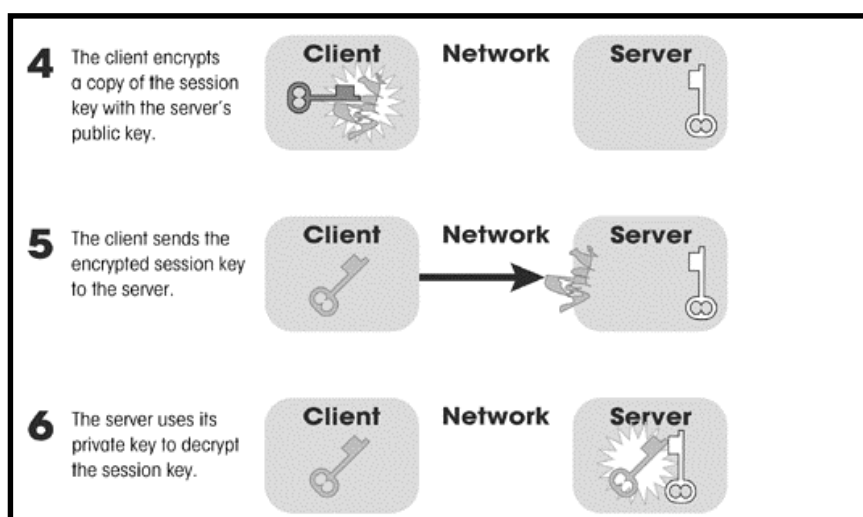
SSL Schlüsselverwendung (I)



WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

21

SSL Schlüsselverwendung (II)



WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

22

SSL Handshake

- SSL Handshake initialisiert Datenaustausch
 - Zertifikatsaustausch
 - Kommunikationspartner belegen ihre Identität
 - Server muss immer ein Zertifikat senden
 - Client-Zertifikat ist optional (mögliche Authentisierung)
 - Aushandlung der Verschlüsselungsmethode
 - Server sendet Liste von Methoden an den Client
 - Client wählt stärkste der vom Server angebotene Methoden
 - Client erzeugt und übermittelt *Session Key*
 - Verschlüsselung mit *Public Key* (sichere Übermittlung)
- Start der Datenübertragung (Record Protocol)
 - Verschlüsselung mit *Session Key* (einfachere Algorithmen)

Grundbegriffe der Sicherheit

- Identifikation (*Identification*)
 - Feststellung der Identität (z.B. Benutzername)
- Authentisierung (*Authentication*)
 - Überprüfung der Identität (z.B. Passwort)
- Autorisierung (*Authorization*)
 - Feststellung der Zugangsberechtigung (Permissions)
- Integrität (*Integrity*)
 - Sicherstellung des korrekten Informationsaustauschs
- Geheimhaltung (*Privacy*)
 - Sicherstellung der geheimen Übertragung

Authentisierung

- Zugriff auf sensible Ressourcen
 - aus Sicherheitsgründen (*Security*)
 - aus Abrechnungsgründen (*Accounting*)
- drei grundlegende Aspekte
 - *Identifikation* (wer will Zugriff?)
 - *Authentisierung* (ist die Identität authentisch?)
 - *Authorisierung* (besteht Zugriffsberechtigung?)
- zwei Wege der Authentisierung (RFC 2617)
 - *Basic Authentication*
 - *Digest Access Authentication*

Basic Authentication

- einfaches Verfahren, von den meisten Browsern unterstützt
- klares Sicherheitsrisiko (unverschlüsselte Übertragung der Informationen)
- Server sendet *401 (unauthorized)* Status und *WWW-Authenticate* Header Field
 - Identifikation eines *authentication schemes*
 - Identifikation eines *authentication realm*
- Client wiederholt Request
 - Benutzer gibt Name und Passwort ein
 - Information in *Authorization* Header Field

Digest Access Authentication

- verschlüsselte Übertragung von Passwort und Name (MD5, RFC 1321)
- Client berechnet *MD5 fingerprint* von
 - Benutzername und Passwort
 - *Realm* (Authentisierungsbereich)
 - *Nonce* (vom Server generiert)
 - HTTP Methode und URI
- Server vergleicht eigenen und erhaltenen *MD5 fingerprint*
- weiterhin bestehende Sicherheitsrisiken
 - initiale Passwortübertragung
 - keine Verschlüsselung des Inhaltes

Zahlungssysteme für E-Commerce

- unabdingbar notwendig für eine komplette Abwicklung von Geschäftsvorgängen
 - reduzieren oder eliminieren Notwendigkeit für einen Medienbruch (je nach Anwendungsfall)
 - schneller, sicherer, effizienter, billiger
- bisher meist Abwicklung über HTML/SSL
 - Verwendung von Kreditkarteninformationen
 - schlechte automatische Weiterverarbeitung
 - Sicherheitsrisiken (hauptsächlich für den Kunden)
- neue Konzepte für Zahlungen notwendig
 - elektronische Kreditkartenbelastungen
 - elektronisches Geld

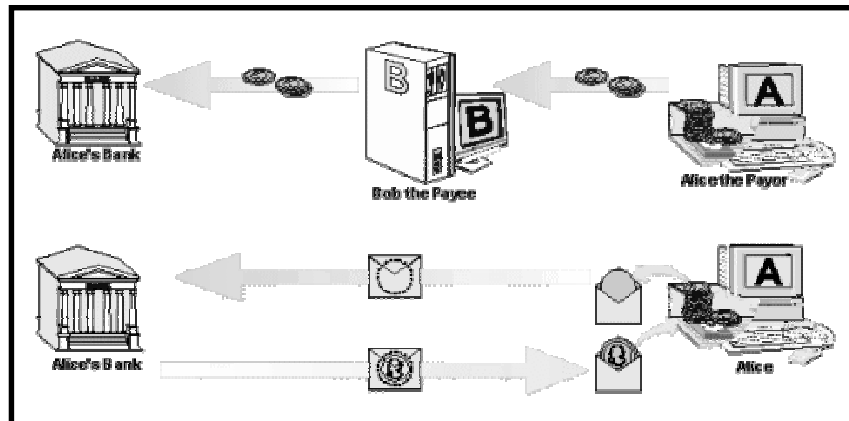
Klassifizierung

- Basierung
 - Konten oder elektronische Münzen?
- Transaktionskonzept
 - Ablauf und beteiligte Parteien einer Transaktion
- Effizienz und Einsatzgebiet
 - Kosten (Eignung für Macro-/Micro-/Pico-Payments)
- Vertraulichkeit und Anonymität
 - Umgang mit und Weitergabe von sensiblen Daten
- Skalierbarkeit
 - zentraler oder verteilter Ansatz?

eCash

- elektronisches Geld
 - anonym (keine rückverfolgbaren Seriennummern)
 - benötigt eine *Purse* auf dem PC des Benutzers
- Probleme bei DigiCash (bald Uebernahme)
- Konto bei einer Bank die eCash ausgibt
 - einige grossen Banken führen eCash Konten
- drei Hauptakteure im eCash Szenario
 - eine Bank, die eCash ausgibt und validiert
 - Kunden mit einem eCash Konto bei der Bank
 - Händler, die eCash akzeptieren
- prinzipielles Problem der Skalierbarkeit

eCash Transaktionen



<http://www.digicash.com/>

WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

31

Secure Electronic Transactions

- Standard für Kreditkartenbelastungen
 - Kooperation von Visa und Mastercard
- Verbesserungen gegenüber gängiger Methode
- sicherer für die Händler
 - keine Käufe mit unautorisierten Kreditkarten
 - keine Widerrufe gegen getätigte Käufe
- sicherer für die Kunden
 - kein Missbrauch der Informationen durch Händler
 - keine Weitergabe der Kreditkartennummer
- sicherer für Banken
 - keine unautorisierten Käufe

WWW (SS2001) - Sicherheitsaspekte, Abschluss und Ausblick

32

Vier SET Komponenten

- *Cardholder Wallet*
 - Software, die bei jedem SET Kunden installiert ist
- *Merchant Server*
 - beim Händler, kommuniziert mit allen anderen Teilen
- *Payment Gateway*
 - Stelle, die Transaktionen zustimmt oder ablehnt
- *Certificate Authority*
 - Ausgabe von Zertifikaten für Kunden und Händler
 - SETCo baut eine eigene CA-Hierarchie auf

Entwicklung der Zahlungssysteme

- keine deutlichen Marktführer erkennbar
- nach wie vor grösstenteils HTML/SSL
- unüberschaubare Menge an Lösungen
 - technische Evaluation zum Teil unmöglich
 - Skalierbarkeit nicht geklärt
- Trend zur Schaffung von *Wallets*
 - Integration verschiedener Mechanismen
 - Methode zur Auswahl eines geeigneten Verfahrens
 - nur eine Lösung des Problems auf Kundenseite
- SET wird sich höchstwahrscheinlich durchsetzen

Zusammenfassung

- Sicherheitsaspekte im Web
 - wichtig aus dem Gesichtspunkt Privacy
 - wichtig aus dem Gesichtspunkt e-Commerce
- Sicherheit an unterschiedlichen Stellen
 - Bestandteil einer Ressource (XML Signature)
 - Eigenschaft einer Verbindung (HTTPS)
 - Eigenschaft einer Transaktion (Zertifikate)
- komplexes Thema mit vielen Variationen
 - zuerst die Grundideen verstehen
 - und dann in Problemfällen anwenden können

"Philosophisches"

- History Repeats itself...
 - Fenstersysteme sind schon alt
 - erste Entwicklungen proprietär (SunView)
 - offene Protokolle für offene Umgebungen (X Windows)
 - Weiterentwicklungen Richtung cleverere Clients
 - Display PostScript bei Sun's NeWS (schnell gestorben)
 - Display PDF als Basis für MacOS X (man wird sehen...)
 - das Web ist auch (für viele...) eine GUI-Plattform
 - am Anfang sehr dummes Interface (HTML+ <ISINDEX>)
 - Weiterentwicklung Richtung mehr Intelligenz
 - DHTML (JavaScript) und HTML Forms, Java
 - weitere Schritte absehbar
 - CC/PP, XForms, XSLT auf dem Client

DA & SA

- mögliche Arbeiten in den Bereichen
 - BibTeXML Weiterentwicklung und Web Site
 - "XLinkbase" Weiterentwicklung Online-Glossary
 - verteilte Topic Maps
 - Kompression von XML ("PER für XML")
- eigene Themen sind durchaus möglich
 - Abschätzung des Aufwandes
 - Forschung in Richtung "related work"
 - Berührungspunkte mit XML/XLink/Topic Maps

Zusammenfassung

- Web als Sammlung von Technologien
 - unübersichtlich und extrem dynamisch
 - Anforderungen von vielen Seiten
 - Anforderungen von unterschiedlichen Seiten
- wichtigste Fähigkeiten als Web-Spezialist
 - Verfolgen der wichtigsten Entwicklungen
 - Erkennen von Parallelen und Gemeinsamkeiten
 - Wissen der möglichen Kombinationen
 - mehr Meta-Wissen als Wissen...

das war's! einen schönen Sommer!