Security & Privacy

Web Architecture and Information Management [./] Spring 2009 — INFO 190-02 (CCN 42509)

Erik Wilde, UC Berkeley School of Information

2009-02-23

Contents

Abstract (2)

TCP and thus HTTP are clear-text protocols, which make no attempt to hide the data being transmitted. For secure data transfers, it thus is necessary to use additional technologies for providing secure data transfers. For the Web, the most interesting security feature are secure HTTP interactions, which are provided by *HTTP over SSL (HTTPS)*, a protocol that layers an encryption layer (SSL or TLS) between TCP and HTTP. For any task involving personalization and/or trust, it is not only necessary to have a concept for providing privacy, but also to have concepts for identity and how to prove identity, which needs authentication.

Security Concepts

Identification (4)

- *Identity* is required for any non-anonymous communications
  - *groups* can have an identity (facebook members see more than non-members)
  - *pseudonyms* are "hidden identities" (the "real identity" is not visible)
  - *personal identity* should be tied to a person itself
- *Proof of Identity* is important for any privileged operation
  - *signatures* and *seals* are traditional ways
  - traditional ways are mostly protected by law (but not really safe)
  - more modern ways often include technicals methods for Authentication [Authentication (1)]
- Client identity on the Web can be bound in three ways:
  1. Computer (most of the time "identified" by an IP Address [Internet Architecture; IP Address (1)])
  2. Browser (in the form of a stored cookie [State Management (Cookies)])
  3. User (identified through some authentication method [Authentication (1)])

Authentication (5)

- *Authentication* is the process of verifying an identity
  - the weakest form of authentication is simply trust
  - legal consequences can make it more risky to falsify authentication
  - technical measures should make it hard to impossible to falsify authenticatio
- Authentication on the Web comes in many different flavors
  - implicitly by accessing a server from some IP Address [Internet Architecture; IP Address (1)] range
  - presenting a cookie [State Management (Cookies)] from a previous formal authentication
  - presenting a password as a proof of identity
  - proving that you are owning additional authentication hardware (often PIN [http://en.wikipedia.org/wiki/Personal_identification_number]-enabled)
- Risk and potential damage should justify authentication methods

Authorization (6)

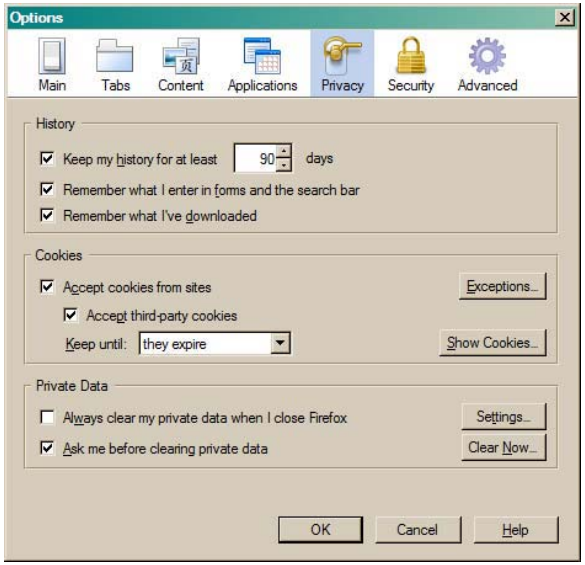
- *Authorization* is the question of allowing operations
  - Identification [Identification (1)] is necessary to identify the initiator
  - Authentication [Authentication (1)] is necessary to verify the initiator's identity
  - if the initiator is authorized, the operation can be performed
- Web pages often are *public* or *restricted access*
  - public web pages do not require any identification (and thus authentication)
  - restricted access Web pages can be group pages (internal company pages)
  - personal access is another popular scenario (email, facebook, online banking)
- Web servers have well-defined ways of performing authentication [Web Foundations (URI and HTTP); HTTP Authentication (1)]

Browser Security & Privacy

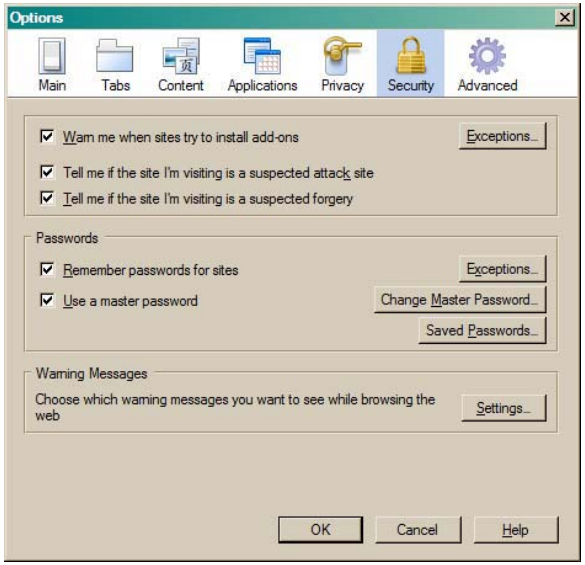
Trust and Security on the Web (8)

- Web-based applications introduce many risks
  - do you trust your browser? (it may not safeguard your information)
  - do you trust your computer? (it may have a virus)
  - do you trust your network? (it may be monitored on various levels)
  - do you trust the server? (it may be a fake phishing [http://en.wikipedia.org/wiki/Phishing] server)
- Most of these risks are amplified by the Web's scale
  - phishing and spamming only work because the Web makes fraud more effective
- Controlling Web access is important for safe browsing
  - trusting shared browsers is risky (they may store logins and cache pages)
  - trusting the network can be risky (more and more networks are wire-tapped)
  - trusting the server is risky (phishing and poor server security)

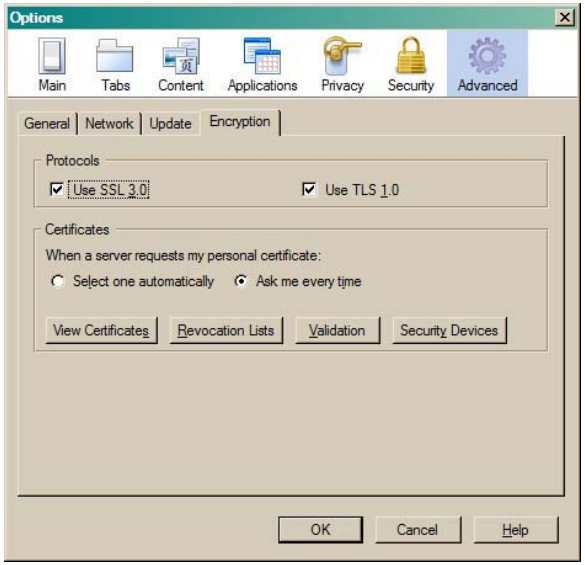
Privacy Options (9)



Security Options (10)



Encryption Options (11)



Security 101

Cryptography (13)

- Cryptography is structured into different layers
  - layering is a well-established principle for *separation of concerns*
- *Cryptographic primitives* implement very basic functionality
  - changes and advancements in this area are limited to very specialized researchers
  - it is easy to make fatal mistakes which then challenge everything built on top if it
- *Cryptographic protocols* assemble primitives into application-level solutions
  - primitives solve very basic security problems (fingerprints, encryption, …)
  - protocols combine these into applications (digital signatures, secure communications, …)

One-Way Function (14)

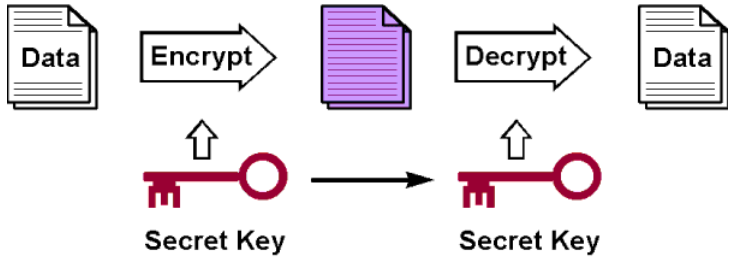
- Hashes (or *message digests*) are well-known in computer science
- One-way functions are cryptographically safe hashes
  - very hard to find an input producing a given output
  - very hard to find two inputs producing the same output ("collision")

Secret-Key Cryptography

Plausible Encryption (16)

- Secret-Key is was most people think of when thinking of encryption
  - *symmetric cryptography* is another popular term
- One key for encryption and decryption
- Losing the key makes encrypted data openly readable
  - there must be a secure channel to transport keys
- Good for long-term relationships with few partners
  - exchange secret keys as part of the initial setup of a relationships
  - adding partners requires a *secure channel* for key exchange
  - changing keys requires a *secure channel* for key exchange
- Almost impractical in an environment with many ad-hoc partners

Notice the Arrow (17)

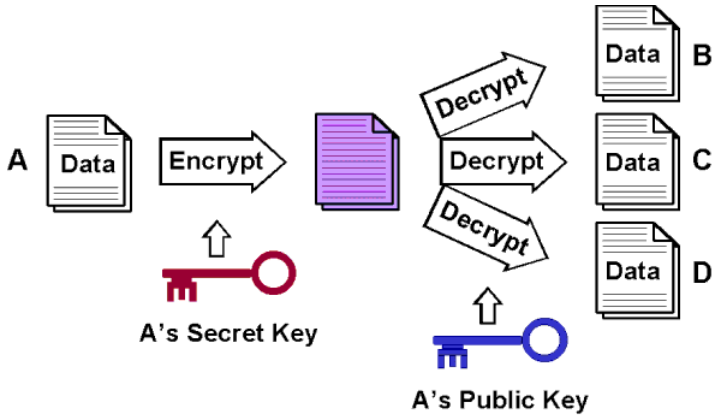


Public-Key Cryptography

Implausible Encryption (19)

- Public-Key intuitively is hard to accept as a concept
  - *asymmetric cryptography* is another popular term
- Key pairs of one public and one secret key
  - *key generation* is the process of generating these key pairs
- The public key can be made available to the public
  - only the secret key can do the inverse operation of the public key
- Good for short-term relationships with many partners
  - publish your public key so that it can be used worldwide
  - everybody can encrypt data using the public key
  - only the owner of the secret can can decrypt the message and read it
- Computationally expensive and not good for a large amounts of data

No Arrow Here … (20)



Cryptographic Protocols

Building Secure Applications (22)

- *Cryptographic primitives* in most cases are not sufficient
  - they provide basic functionality for fundamental tasks
  - they must by combined to provide solutions for real-world problems
- Typical problem #1: How to ensure key authenticity
  - with insecure keys, the majority of cryptographic methods is worthless
- Typical problem #2: How to communicate securely without shared keys
  - many interesting scenarios are based on ad-hoc interactions
  - secret-key does not work, public-key needs to verify the peer
- Typical problem #3: How to check authenticity and integrity of data
  - integrity can be done with checksums, but these could be forged
  - authenticity needs a cryptographically secure way of combining identity and data

Certificate (23)

- Certificates are digital signatures issued by a trusted party
  - most digital signatures are created with certified public keys
  - this means the digital signature is created based on a digitally signed key
- Who can you trust on the Web?
  - trust can only start to grow based on initial trust in something
  - many systems come with pre-installed trust (*root certificates*)
  - certificates from other issuers will cause browsers to complain [https://katapultmedia.com/]
- Certificates (like domain names) are a very easy way to make money
  - in theory there are different levels of certificates with different levels of identity checking
  - in practice most sites choose the cheapest one that does not give an error message

HTTP over SSL (HTTPS)

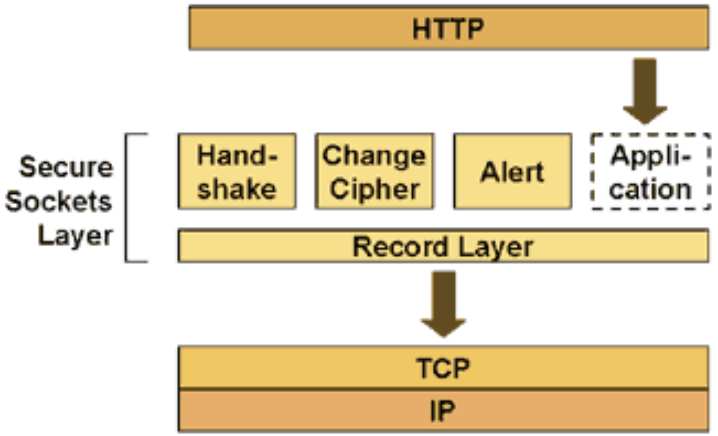
Secure Communications (25)

- Public-Key cryptography [Public-Key Cryptography (1)] is computationally expensive
  - it is possible to encrypt all traffic using asymmetric key pairs
  - this generates considerably more load on the server side
- Combining public-key [Public-Key Cryptography (1)] and secret-key [Secret-Key Cryptography (1)] cryptography
  1. check the public key for authenticity (using a Certificate [Certificate (1)])
  2. generate a key for a secret-key encryption scheme
  3. use the public key to securely transmit the secret key
  4. use the secret key for securely transmitting the payload
- Combines the advantages of both methods
  - the lower complexity of secret-key algorithms
  - the ability of public-key algorithms to work without a secure channel

HTTP and Security (26)

- HTTP sends clear-text messages
- Making HTTP secure requires additional mechanisms
- Encryption is done by a layer on top of TCP
  - *Secure Sockets Layer (SSL)* is the protocol layer invented by Netscape
  - *Transport Layer Security (TLS)* is the standardized Internet version
  - TLS adds more encryption schemes and more flexibility
- Lower-level methods may also provide encryption
  - *Virtual Private Networks (VPN)* provide IP-based encryption
  - *WEP* and *WPA* provide network interface encryption

HTTP and SSL (27)



Conclusions

Internet Security (29)

- Certificates are used to guarantee a party's authenticity
- Certificates are digital signatures issued by trusted parties
- One authenticated, public keys can be used to securely communicate
- Encryption on the Web is based on HTTPS