



Certificate-less User Authentication with Consent

Workshop on Digital Identity Management (DIM) 2007

Shingo Orihara, Yukio Tsuruoka, and Kenji Takahashi
NTT Information Sharing Platform Laboratories



Outline

- New user authentication scheme obtained by combining password and public key cryptography
- Implementation using new HTML tags



User authentication at Web sites

◆ Do we have a secure and highly usable authentication scheme?

■ Passwords

- Widely used
- Easily stolen
- Difficult to remember

■ SSL client authentication

- Secret key is not revealed
- Not widely used
- Difficult to handle user certificates



What is really needed?

Most electronic commerce sites want to...

1. provide personalized services for each user,
Introduce user account and distinguish users.
2. prevent ID theft and impersonation,
Strong authentication, e.g., public key scheme.
3. let users indicate their intention to log in,
By *what you know*, e.g., password.
4. manage accounting & payment.
Use existing platform, e.g., credit card.

We focus on 1, 2, and 3.

For most EC sites, the important thing is...
**recognizing a user as the same user
at his/her second visit and later.**

Continuity of users' identities

This is enough to provide personalized services
for users.

It is not important who the user really is.

EC sites trust the personal information the user
provides, such as name and address.

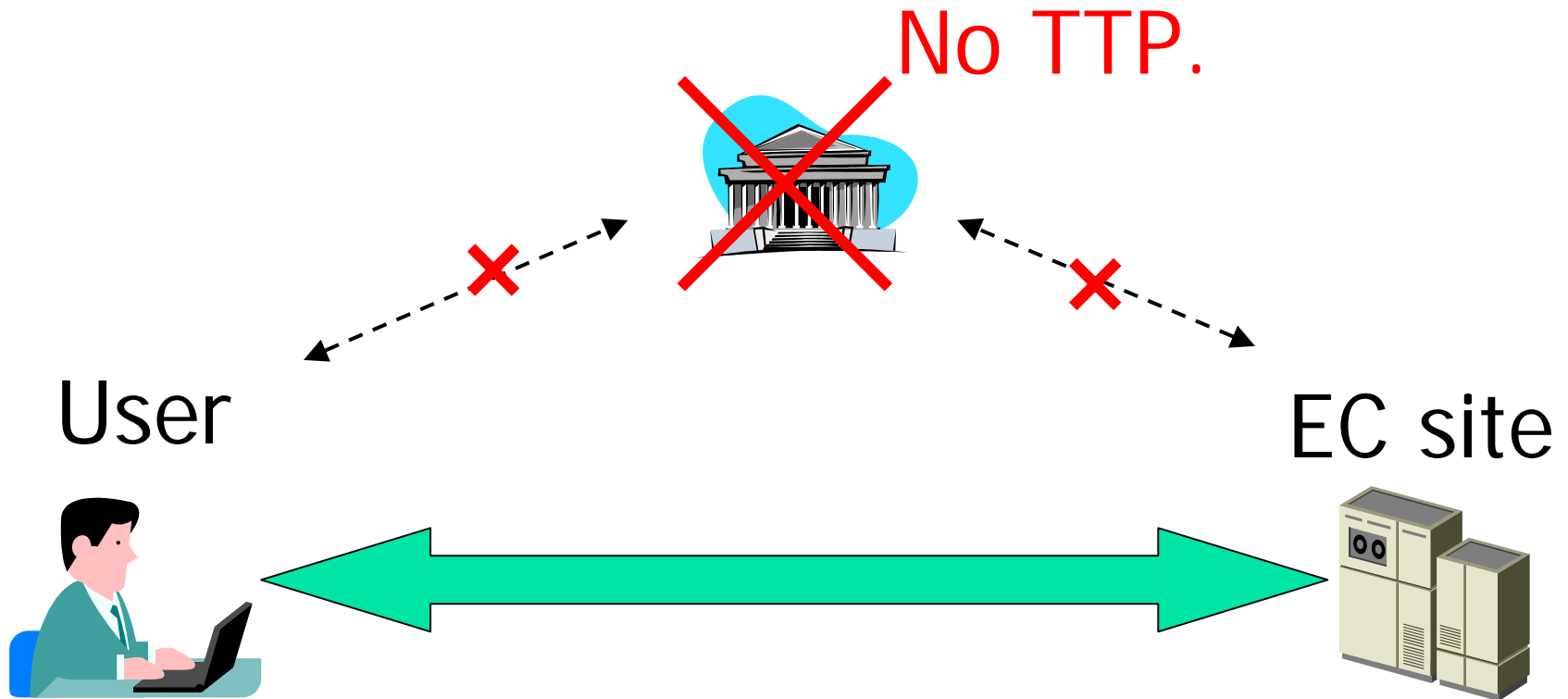
No personal reference by TTP is required



Our scheme

- Assures continuity of users' identities.
- No user certificate is required.
- No TTP, e.g., CA, is required.
- Combination of password authentication and public key scheme
 - PW: for indicating users' intention
 - PK: for strong authentication
- Using public key scheme overcomes weakness of password authentication.

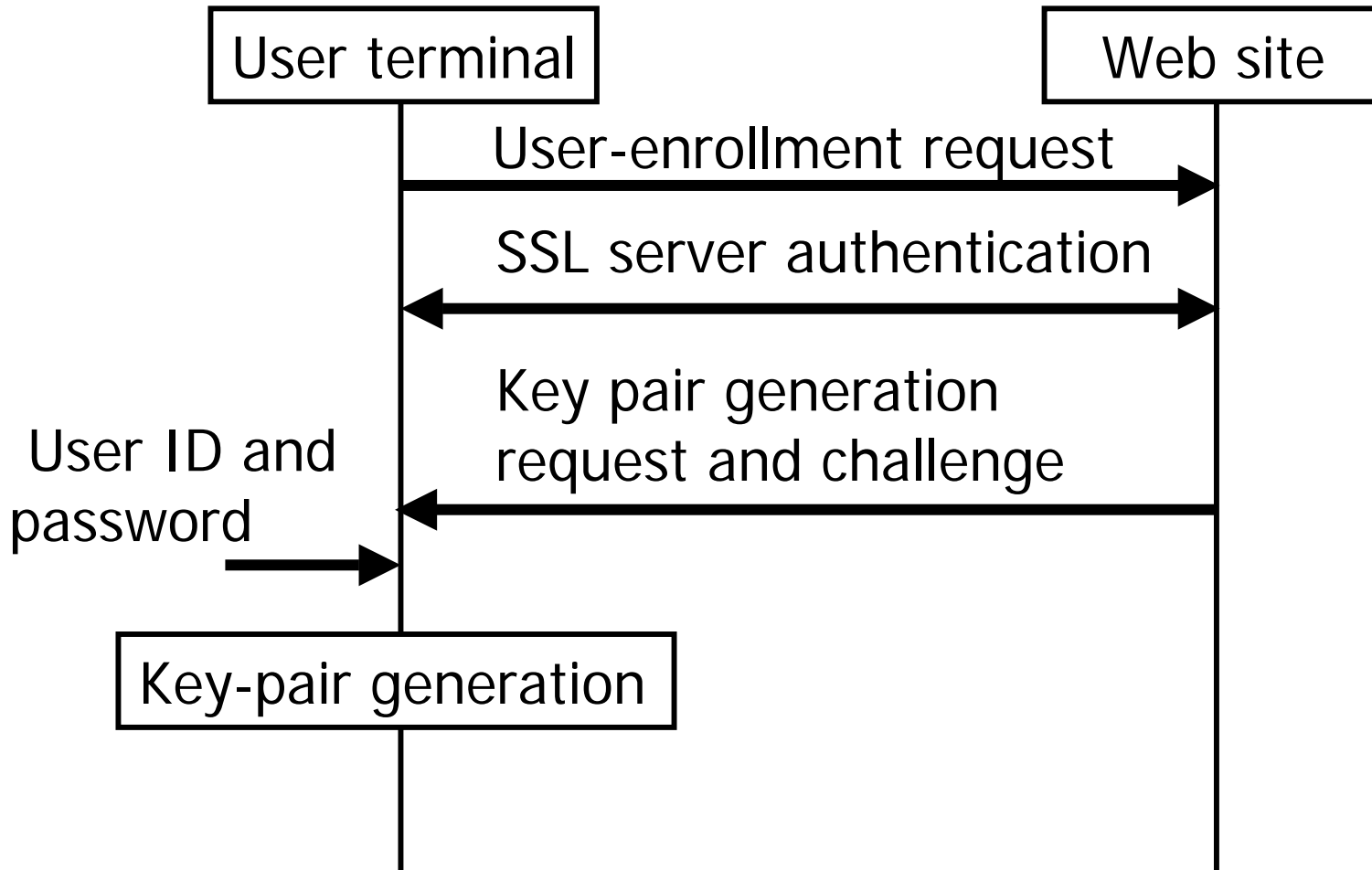
Our scheme (cont.)



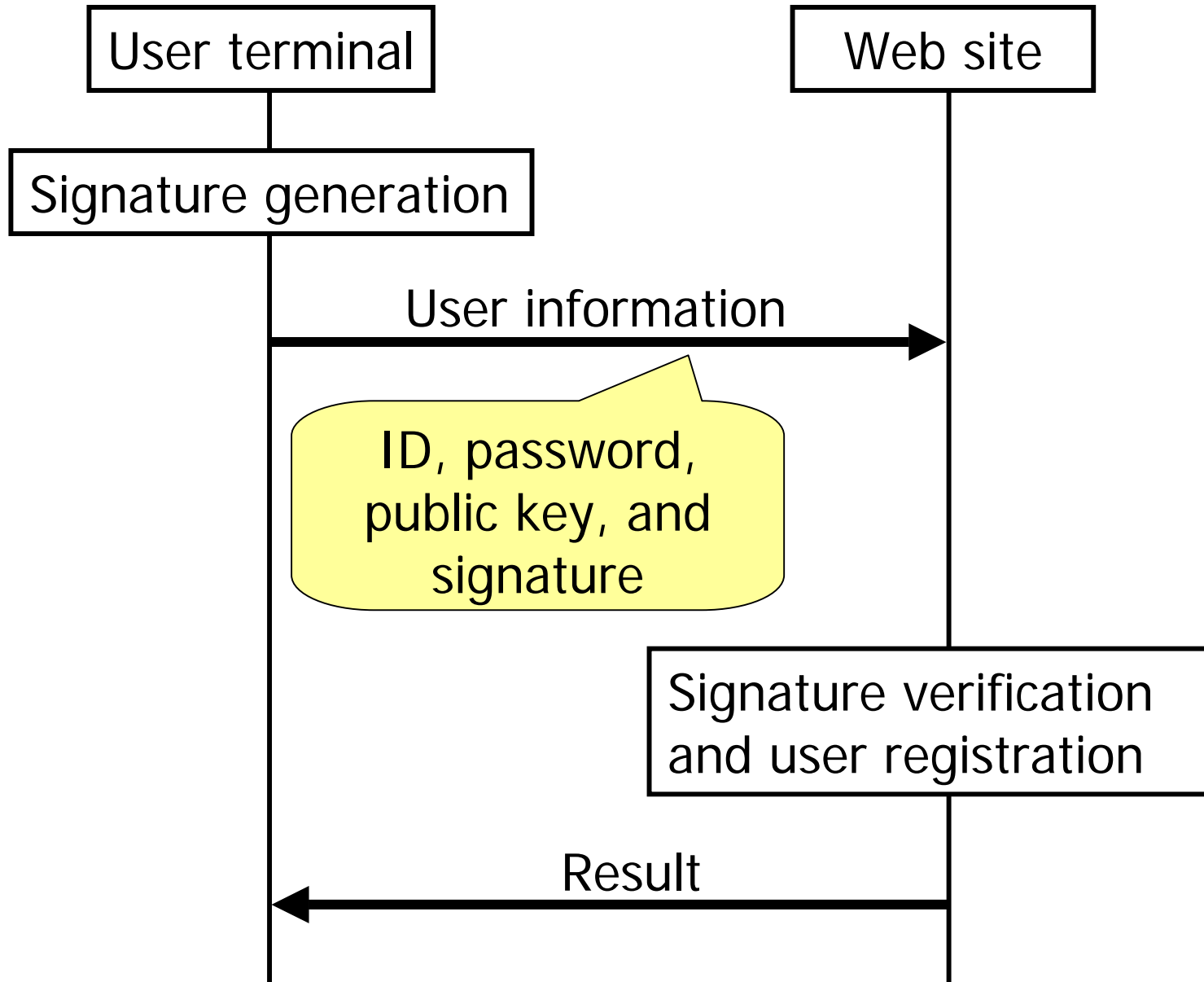
Communication occurs between a user and an EC site (2-party protocol).

How it works

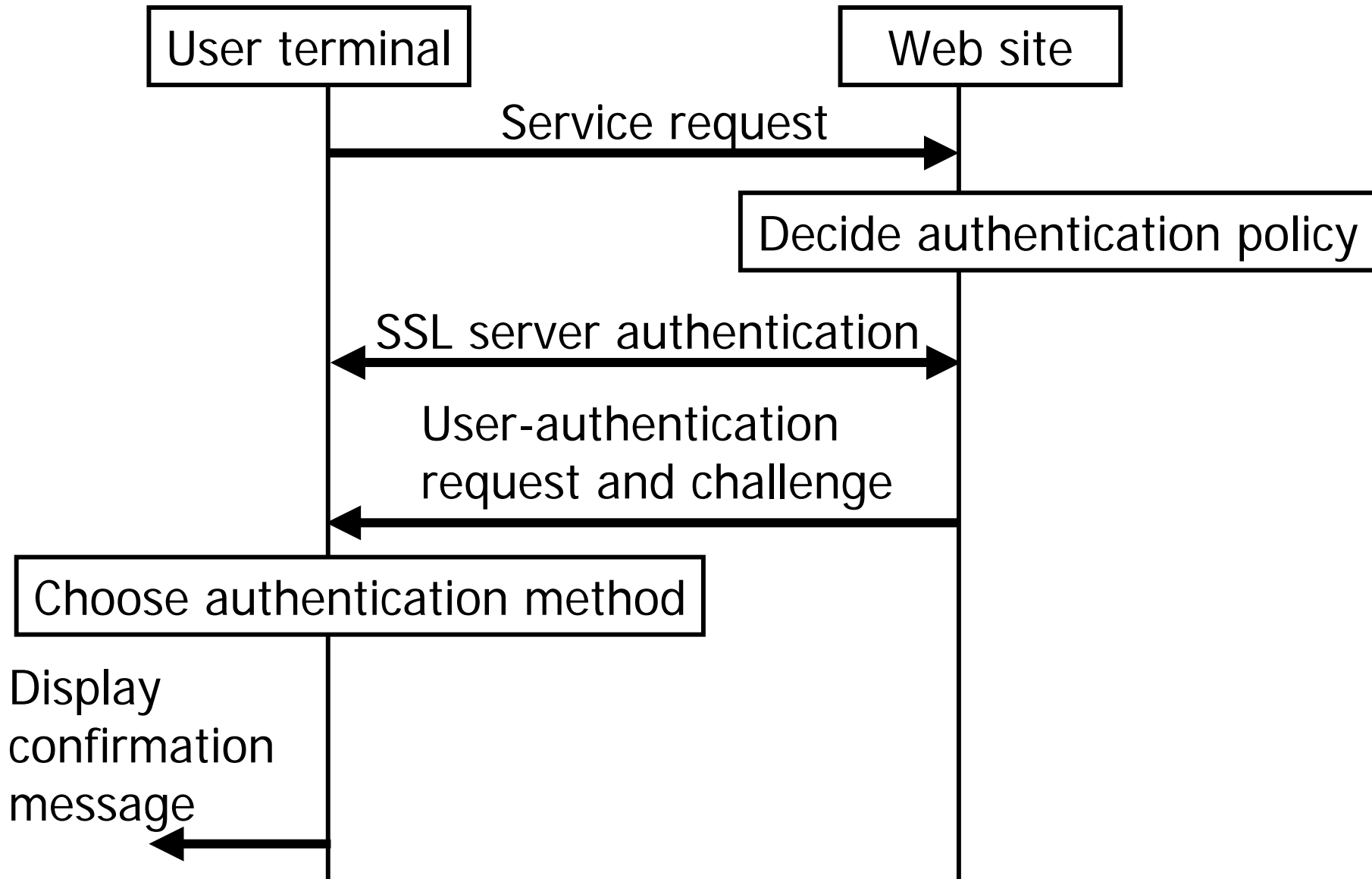
User enrollment phase



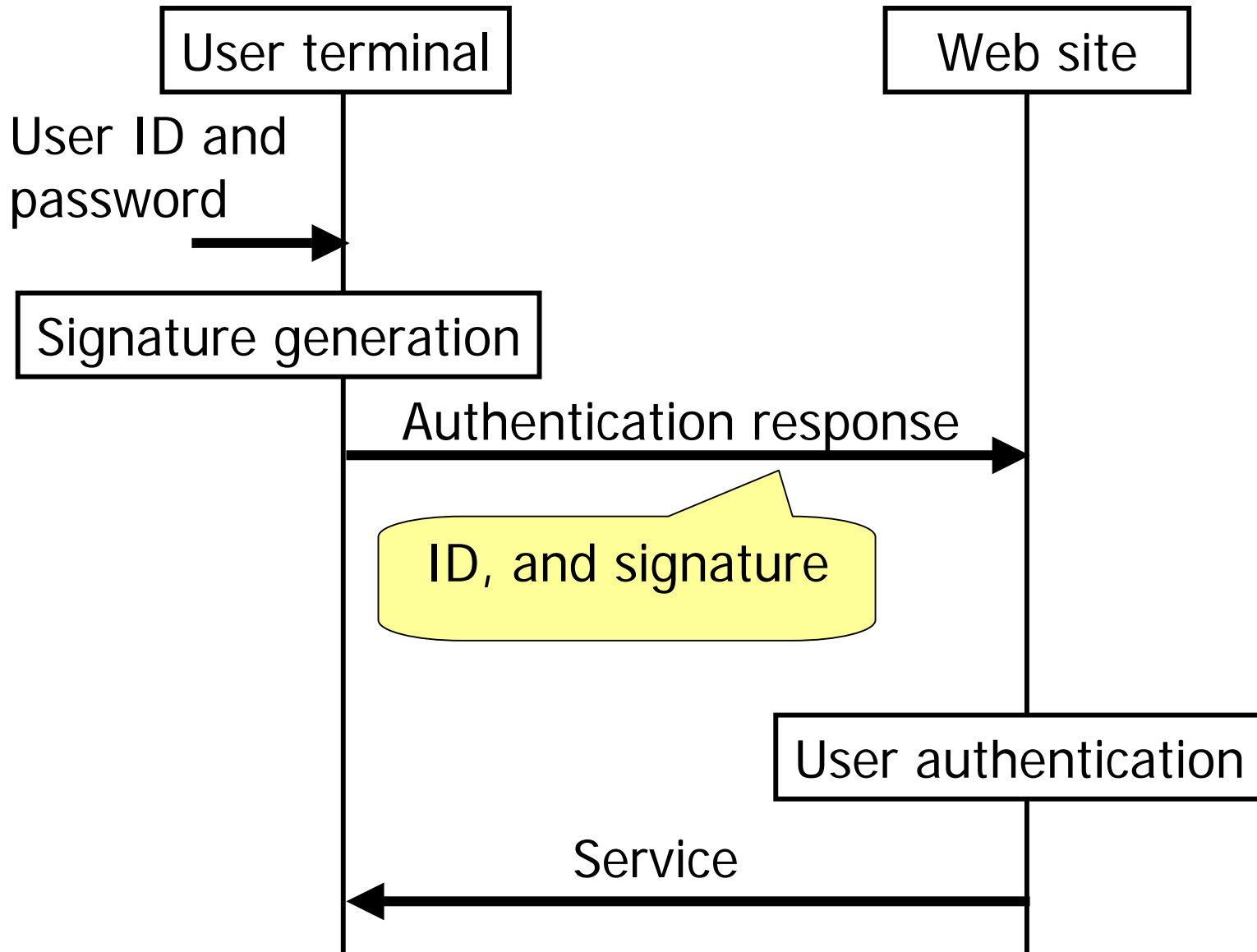
User enrollment phase (cont.)



User authentication phase



User authentication phase (cont.)





Design in detail

◆ Introduce two new HTML tags.

➤ **KEYREG**

- For public key registration
- Embedded in a user registration form

➤ **AUTH**

- For authentication
- Embedded in an ID/password form

New tag: KEYREG

- Key pair generation and signature calculation
- Usage:

```
<form action="registration.php" method="post" >  
  <label>Name:</label><input type="text" name="name">  
  <label>ID:</label><input type="text" name="id">  
  <label>Password:</label><input type="password" name="pw">  
  <KEYREG name="pk" challenge="0123abcdef"  
    ...other attributes, e.g., policy>  
  <input type="submit" value="Send">  
</form>
```

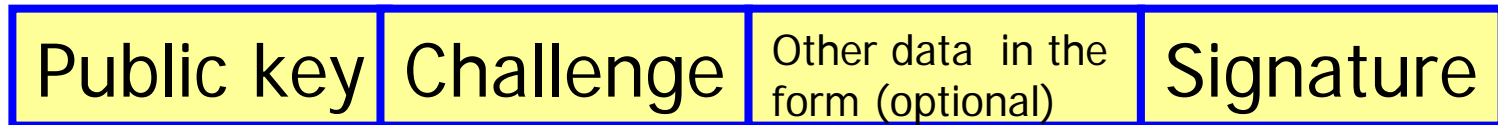
Embedded in a user
registration form

How KEYREG works

When the submit button is clicked,

1. Generate private and public key pair, like Netscape and Firefox do for keygen tag.
2. Calculate response by signing the public key and the challenge string with the private key.

Response:



Data to be signed.

3. Post the response with other data in the form.

New tag: AUTH

- Perform challenge-response authentication
- Usage:

```
<form action="login.php" method="post" >  
  <label>ID:</label> <input type="text" name="id">  
  <label>Password:</label> <input type="password" name="pw">  
  <AUTH name="pk" challenge="xyz9876"  
  ...other attributes, e.g., policy>  
  <input type="submit" value="Send">  
</form>
```

Embedded in an
ID/password form

How AUTH works

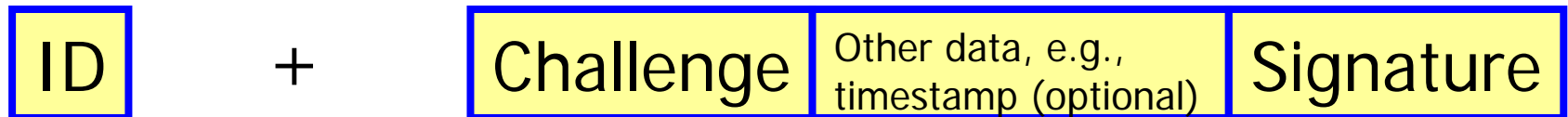
When the submit button is clicked,

1. Calculate response by signing the ID, the password, the challenge string, and other optional data with the private key.

Data to be signed:



Response:



3. Post the response with other data in the form.



Modification at Web sites

- User database to register users' public keys
- Insert KEYREG tag into a user registration page
- Insert AUTH tag into a login page
- Prepare CGIs to handle KEYREG and AUTH
(e.g., public key registration and signature verification)

Not much modification is required.



Unadapted browsers

```
<form action="registration.php" method="post" >
  <label>Name:</label><input type="text" name="name">
  <label>ID:</label><input type="text" name="id">
  <label>Password:</label><input type="password" name="pw">
  <KEYREG name="pk" challenge="0123abcdef"
    ...other attributes, e.g., policy>
  <input type="submit" value="Send">
</form>
```

KEYREG is ignored and a usual user registration form appears.

AUTH is ignored similarly, a usual ID/password form appears.



Policy handling

Our scheme basically uses a password and public key simultaneously.

In some circumstances, only a password or public key is enough to authenticate a user.

By introducing authentication policies, we enable clients to select an appropriate authentication method.

We consider 5 authentication methods.

- **NA** (no action): User does not have to do anything. Login is done automatically and anonymously.
- **OK**: User must click on an OK button. (e.g., agreement with the terms and conditions.) This is used when an indication of user's intention is needed.
- **PW** (password): User must enter his/her password.
- **PK** (public key): User must sign a response message with his/her private key. This is usually done automatically by browser program.
- **PKPW** (public key and password): User must enter his/her password and sign it.

As an example of policies, we defined 6 policies by combining of **required security level** and **indication of user's intention** as follows.

Indication of user's intention	Security level		
	0 (low)	1 (medium)	2 (high)
0 (not required)	NA, OK, PW, PK, PKPW	PW, PK, PKPW	PK, PKPW
1 (required)	OK, PW, PKPW	PW, PKPW	PKPW

For each policy, there are appropriate authentication methods, as shown in the table.

User terminal

Web site

<AUTH ... policy="high, required" ...>

Select an authentication method suitable for the given policy.

Ask user to input ID and password if needed.

Make response.

Report selected method, e.g., PKPW, and send response.

Authenticate the user by selected method.



Advantages of our scheme

Compared with typical password authentication

- Signed passwords can be a proof of valid login later.

Compared with typical authentication with PKI

- By asking a user to enter a password, sites can confirm the user's intention to log in.



Conclusion

- New user authentication scheme
- Combination of password authentication and public key cryptography overcomes weakness of password authentication
- Two new HTML tags, for key pair generation and signature calculation



Future work

- Implementation and evaluation of the scheme
- More consideration about key life-cycle management, e.g., deletion and revocation