Notes: Privacy Considerations for Capture-Free Throttling

Markus Jakobsson Elaine Shi

January 6, 2009

1 Privacy Goals

- The trusted authority should not learn which clients visited which site.
- The trusted authority should not learn how many users or the volumn of traffic at each site, as the service/site providers may consider this information as business secret.

2 Proposed Scheme

2.1 Notations

notation	meaning
T	trusted authority.
S	service/site provider.
C	client.
usk	user secret key of the group signature scheme
$\mathsf{gsig}_{\mathsf{usk}}(msg)$	group signature on user secret key usk and message msg.
sk_T	secret key of T

2.2 Protocol

Setup. The client C sends his/her authentication information (e.g., credit card number, using sms messages) to the trusted authority T. T plants a cache cookie denoted `cookie.html` in C's browser. `cookie.html` contains a secret user key usk which will later be used to sign a group signature.

Authentication. On a high level, the client C requests a page from S. S redirects C to T. After authenticating itself to T, C obtains a signature from T. C forwards this signature to S, who verifies the signature and grants C access. All of the above happens tranparent to the end user.

1. $C \rightarrow S$: request for `http://S/landing_page`
2. $S \rightarrow C$: landing page containing an iframe to `http://T/nonce`
3. $C \rightarrow T$: request for `http://T/nonce`

4. $T \rightarrow C : T$ returns C a page P. This page imports `http://T/cookie.html` e.g., by including it as source: <`src=http://T/cookie.html`>. In addition, the page P contains the javascript code for computing group signature.

5. C : The javascript code on page P reads the user credential usk embedded in the cache cookie `cookie.html`, and computes:
$$\mathsf{gsig}_{\mathsf{usk}}(t)$$
where t denotes the current time epoch.

6. $C \rightarrow T$: The javascript code sends $\mathsf{gsig}_{\mathsf{usk}}(t)$ to T.

7. T: verify the group signature.

8. $T \rightarrow C$: $\sigma_{\mathsf{sk}_T}(\texttt{nonce})$

9. $C \rightarrow S$: `http://S/nonce+`$\sigma_{\mathsf{sk}_T}(\texttt{nonce})$

10. S: Verify $\sigma_{\mathsf{sk}_T}(\texttt{nonce})$.

2.3 Throttling with Group Signatures

As group signatures protect the anonymity of the signer, how can T enforce throttling for each user?

We can use *one-show tags* to solve this problem. A one-show tag is a group signature scheme with an additional one-show function F. Given any valid group signature σ on a user secret key usk and a message msg, $F(\sigma)$ yields a deterministic outcome. Conceptually, one can think of a one-show tag scheme as a deterministic group signature scheme. As long as the user signs each message only once, the user's anonymity is still protected. However, if a user signs the same messages twice, then the two signatures can be linked, because the signature is deterministic.

Assume we allow k connections per user for each time epoch. We need to change the group signature in the above description as below.

Let t denote the current time epoch. The client picks a random $r \in [1, k]$ which has not been used in the current time epoch, and computes the following group signature:

$$\mathsf{gsig}_{\mathsf{usk}}(r, t)$$

Discussions. One question is how the client can pick an unused r if the client has access to multiple machines. This could be solved at setup. The client has to log in to T at each machine. The server can embed a machine identifier i in the cache cookie. Now the client signs the tuple (i, r, t) using the one-show tag scheme. A client having m machines gets $m \cdot k$ accesses per time epoch. The trusted authority can prevent a client from registering too many machines to prevent malware or viruses.