Fraud Vulnerabilities in SiteKey Security at Bank of America

Jim Youll <jim@cr-labs.com>
CTO, Challenge/Response, LLC
Cambridge, MA

Review draft to Bank of America/RSA: June 26, 2006
This version: July 18, 2006

Abstract

The SiteKey anti-phishing system used by Bank of America and other financial institutions is susceptible to a real-time exploit in which an attacker can create a fake web page that includes a victim's correct, secret SiteKey image, text phrase and challenge questions. This paper discusses the customer-facing implementation of SiteKey as seen from a web browser, the reasons for its vulnerabilities, the risks posed by its design and by its persistent storage of a security-weakening token, and the means by which those vulnerabilities could be exploited. Possible improvements are proposed, though the accompanying discussion argues that the single-ended authentication used by SiteKey and other systems is not a sufficient deterrent to phishing or other online frauds. Also included is a brief summary of a discussion between the author and representatives of Bank of America and RSA Security regarding the paper and the bank's overall approach to customer safety and security. This report does not provide source code or detailed instructions about carrying out the described attacks.

Summary

Persistent, large-scale phishing attacks have disrupted the flow of information between financial institutions and their customers. In a phishing attack, a scammer creates a fake version of a web site, then lures victims to it with authentic looking e-mails. The sole purpose of the fake site is to trick victims into entering their secrets – user names and passwords. Attackers sell the captured secrets or use them to steal directly from their victims.

SiteKey is an anti-phishing system recently adopted by Bank of America and other financial institutions. A product of PassMark Security (now part of RSA Security, which is itself under acquisition by EMC Inc.), SiteKey is said to foil phishing by proving the authenticity of protected web sites. SiteKey is also said to prevent unauthorized access to an account even when the user name and password have been stolen, by asking personal questions with difficult-to-guess answers such as "What was the name of your first pet?"

SiteKey equipped web sites establish a "shared secret" – a picture and a text phrase – between the bank and the customer. When the customer starts to log in, the web site displays these items to "prove" it is legitimate. Further, if the site has not been previously accessed from a given computer, it will present one of three "challenge questions" whose answers are known only to the bank and the customer. The rationale for this approach is that a web site that shows a correct "shared secret" must be real: a bad guy could not discover or guess a text phrase and picture that were set up privately between a bank and a customer. Nor could a bad guy guess personal information such as the name of a customer's first pet.

Unfortunately, SiteKey as deployed by Bank of America, and probably as deployed by others, does not provide appreciable protection from typical phishing scams. A scammer cannot guess the secret image and phrase, but these items can be obtained in real time with the unwitting assistance of a victim and his or her bank.

Finding 1

SiteKey is susceptible to a real-time, man-in-the-middle attack. An attacker can create a fake web site that looks like a legitimate Bank of America web site, including a victim's correct SiteKey secret image and text phrase.

A SiteKey shielded site is slightly more difficult to fake than others. The key difference is that an attacker must create a "proxy" to relay a small amount of information between the victim (viewing the fake site) and the legitimate web server. The scripts for a server or bot-infected computer[1] to carry this out would be simple to create and are well within the means of an attacker seeking a financial payoff. Complicating matters, Bank of America promotes extreme confidence in SiteKey that could unintentionally persuade an otherwise skeptical customer that a fake site is real.

Finding 2

On every computer from which a customer accesses a BofA account, SiteKey may store a token that permanently bypasses the "challenge questions" and permits account access with just a user name and password. The bypass token is held in persistent storage on each computer, can be replayed, is long-lived, and may be copied from one computer to another. There is no easy means to "de-authorize" a computer or invalidate a token once it has been created, potentially creating long-term, invisible vulnerabilities.

Comments from Bank of America and RSA Security[2]

The author has spoken with representatives of Bank of America and RSA Security who reviewed a draft of this report. The gist of their response is that SiteKey is one of many components – some visible outside the bank, some not – that comprise a holistic strategy against phishing and other crimes. The representatives said that SiteKey is helpful in revealing and monitoring real-time attacks of the type described here, that real-time attacks are easier to monitor than attacks against static web pages, and that the bank uses SiteKey to thwart large-scale phishing that might injure many customers at once. Bank of America also offers "$0 liability guarantee" [3] to online customers in the event of a loss despite these security measures. The representatives' sense of this report is that it is accurate but misleading in that it addresses a risk in an isolated element of a larger security apparatus.

[1] http://en.wikipedia.org/wiki/Zombie_computer

[2] The author does not intend to speak for these individuals. This summary is included in the interest of fairness and openness.

[3] http://www.bankofamerica.com/privacy/index.cfm?template=privacysecur_olb

Notes

- This report considers SiteKey at Bank of America Corporation, but the general concerns raised probably apply to other SiteKey installations. These same concerns are also generally applicable to any security method that relies solely on server-side security to detect and stop online frauds.

- The information needed to understand SiteKey – or to perpetrate phishing frauds against its users – can be acquired with little more than a web browser and optionally a simple HTTP or TCP monitor such as tcpdump,[4] or from the traffic logs of a web proxy such as Squid.[5]

- No privileged access to any server or other device is required to discover the workings of SiteKey or to build or operate a man-in-the-middle proxy as described here.

- It is not necessary to decrypt, reverse-engineer or otherwise "crack" the encryption or other security mechanisms of the SiteKey protocol to carry out the attack described here. The ability to proxy and rewrite traffic between a victim and a legitimate server is sufficient.

- Example data have been obfuscated or truncated. For example, cookie and token values have been changed from observed values. These changes do not impact the processes discussed here, or the conclusions drawn.

- Examples have been abbreviated for clarity and do not always show step by step detail.

- SiteKey is a registered service mark of Bank of America Corporation.

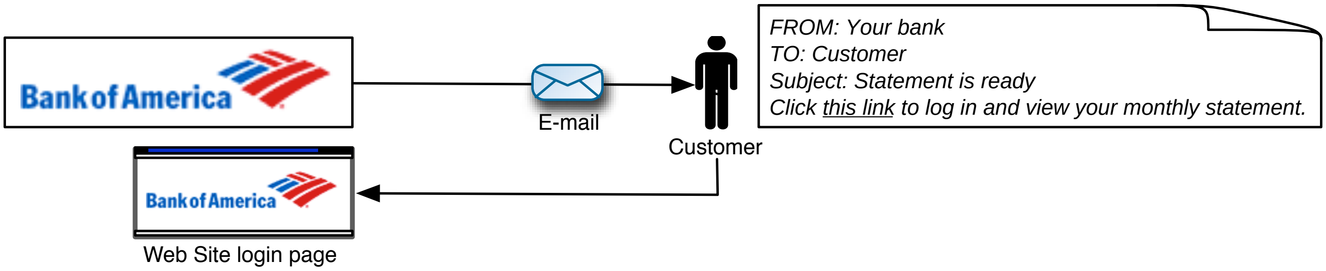
[4] http://www.ethereal.com/docs/man-pages/tcpdump.8.html

[5] http://www.squid-cache.org/

Background

Phishing frauds

Phishing attacks have frustrated efforts by financial institutions such as banks, brokerages and credit unions to communicate with their customers through the Internet. Of particular concern are electronic mail messages that link to web pages holding account information such as monthly statements or trade confirmations. The useful, familiar and intuitive email-to-web path (Fig. 1) is the chief target of phishing attacks, as fraudsters attempt to lure victims to fake financial web sites, in order to steal their access secrets.



FROM: Your bank
TO: Customer
Subject: Statement is ready
Click this link to log in and view your monthly statement.

Fig. 1: The legitimate email-to-web cycle

The ongoing flood of phishing e-mail messages is confusing and dangerous to consumers and financial institutions. Sorting real messages from fakes is not always easy. For every legitimate message received from a bank or credit union, a customer may have to dodge dozens or even hundreds of fakes that would lead them to fraudulent web sites. The confusion can be so great and the frauds so well done, that even security officers of phishing targets have insistently mistaken fake messages and web sites for the real thing.[6,7]

Misidentifying a single bogus message as a real one may bring painful consequences ranging from lost money to full blown identity theft. Those who are not victimized may even become reluctant to trust online transactions, increasing the cost and frustration around services that were intended to reduce expenses and please customers.

Bank of America's SiteKey implementation

Charlotte, North Carolina-based Bank of America ("BofA") began the transition to a new login protocol for its 14.7 million online banking customers[8] in mid-2005, choosing the "SiteKey" product by Menlo Park, California-based PassMark Security.[9] Passmark Security was acquired by RSA Security in April, 2006.[10] As of July 2006, RSA is in the process of acquisition by EMC, Inc.[11]

SiteKey aims to increase security and customer confidence in two ways: (1) from bank to customer, reassuring customers that they are entering their user names and passwords into a real BofA web site; (2) from customer to bank, thwarting unauthorized access to accounts even if a user name and password are compromised, by asking "challenge questions" such as "What was the name of your first pet?"

A May 2005 press release said SiteKey would help customers...

> *"... confirm the Web site's validity. When customers log in to Online Banking ... they can click on the SiteKey button to see their secret image and phrase. If the image and phrase don't appear, the customer could be at a fraudulent site. This capability targets a fraud called spoofing, and will help ensure customers are not fooled by fake versions of bankofamerica.com."* [12]

[6] Ebay tricked by phony e-mail, by Robert McMillan, InfoWorld, Dec 5, 2005

[7] Richi Jennings Blog, Dec 2005, http://richi.co.uk/blog/2005/12/ebays-anti-phishing-desk-sucks.html

[8] Bank of America Full Year 2005 Investor Fact Book, http://investor.bankofamerica.com/

[9] http://www.passmarksecurity.com/

[10] RSA Buys Consumer Authentication Specialist, EWeek.com, Apr 24, 2006 http://www.eweek.com/article2/0,1759,1952966,00.asp

[11] EMC Acquires RSA, Paul Shread, InternetNews,com, http://www.internetnews.com/storage/article.php/3617376

[12] Bank of America announces industry-leading security feature for its 13.2 million online banking customers to help prevent fraud and identity theft, May 26, 2005 http://newsroom.bankofamerica.com/index.php?s=press_releases&item=6971

BofA debuted SiteKey in the Tennessee market area in June 2005 and had planned to roll it out nationwide by the end of the year. The rollout was delayed in the fourth quarter of 2005 for unspecified adjustments,[13] but resumed by Jan. 2006. On June 12, 2006, BofA announced that the system was active in all markets, and was now a standard component of all electronic banking logins.[14]

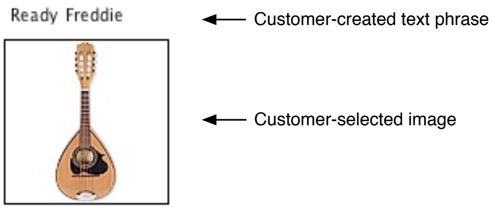


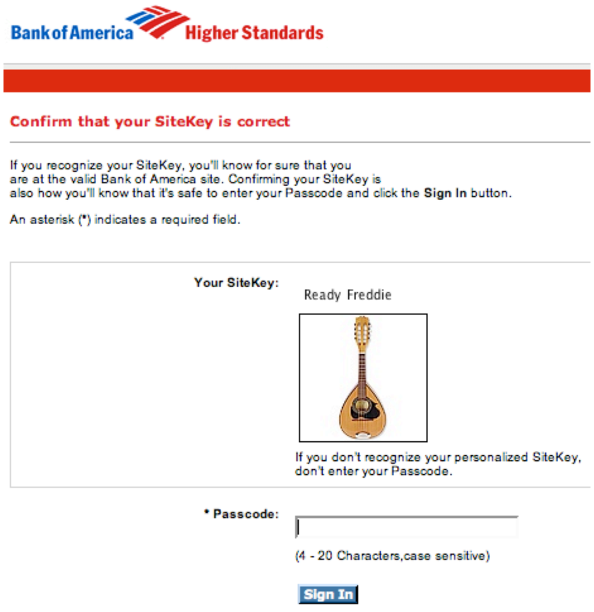
Fig. 2: A user-selected SiteKey image and user-created text phrase

The bank-to-customer security feature is easy to understand and use: during the login process, but before the customer has entered a password, SiteKey displays a shared secret consisting of a picture and text phrase (Fig. 2,3). These are configured by the customer and known only to the bank and the customer. If the image and phrase shown are correct, customers are told to be fully confident that they are connected to a real BofA web site. If the image and phrase are absent or incorrect, the site may be fraudulent and the customer should stop immediately and contact the bank. To reinforce awareness of this feature, BofA frequently reminds customers to look for the SiteKey image and phrase when logging in.

The bank-to-customer component of SiteKey security has two elements: (1) “proof of identity” consisting of the image and a text phrase that are shown together on the password-entry page; (2) validation consisting of the customer’s review and confirmation of the proof of identity.

Customers design the “proof of identity” during their initial enrollment by choosing an image from a bank-provided collection of pictures, and typing a free-form text phrase. At enrollment time, the customer also selects three “challenge questions” and enters their answers.

Once enrolled, a customer may change the image, text phrase or challenge questions and answers at any time. This sort of authentication with shared secrets is analogous to a customer’s entering a password (a secret that only the customer and the bank know) but running in reverse, with the bank showing a secret to the customer.



Confirm that your SiteKey is correct

If you recognize your SiteKey, you'll know for sure that you are at the valid Bank of America site. Confirming your SiteKey is also how you'll know that it's safe to enter your Passcode and click the **Sign In** button.

An asterisk (*) indicates a required field.

Your SiteKey: Ready Freddie

If you don't recognize your personalized SiteKey, don't enter your Passcode.

*** Passcode:**

(4 - 20 Characters,case sensitive)

Fig. 3: SiteKey image and text phrase on the “enter Passcode” page

[13] Bank of America's move to stronger authentication delayed, ComputerWorld, Oct 21, 2005 http://www.computerworld.com/securitytopics/security/story/0,10801,105620,00.html?source=x73

[14] Bank of America Launches Site Key™ Online Banking Security Service in Washington and Idaho, Jun 12, 2006 http://newsroom.bankofamerica.com/index.php?s=press_releases&item=7450

How SiteKey works: from the web browser's perspective

SiteKey adds new authentication steps to the traditional user name/password login process. If the customer is connecting from a machine not previously used for online banking, SiteKey will pose a "challenge question" to further confirm the customer's identity (Fig. 4). If the answer is correct, SiteKey will "remember" the computer by placing a long-lived token (similar to a cookie) on the computer, in an Adobe Flash "persistent shared object." If the machine was previously introduced to SiteKey, the machine sends its token to the SiteKey server at step 2, and no challenge question is asked (see Fig. 5).

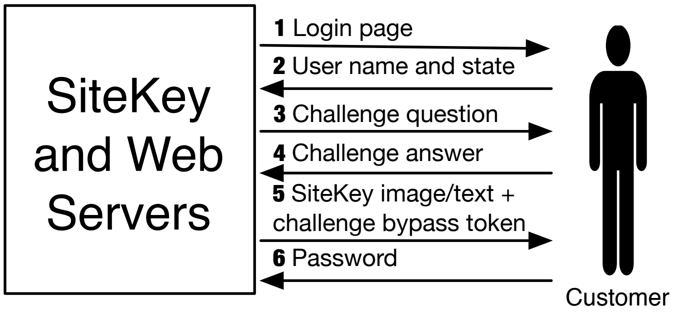


Fig. 4: SiteKey login from an unfamiliar computer, including a challenge question and answer

Description of the full SiteKey login process illustrated in Fig. 4

1. The customer opens the bank's web site, which could use HTTPS (secure, or "SSL"), or regular HTTP.
2. On the login page (typically the home page) the customer enters a user name (BofA calls this "Online ID"). The customer *does not* enter a password here, but may enter additional qualifying information, such as the state where the account is located. The customer then clicks "Sign in."

   SiteKey checks cookies and the Flash persistent shared object for a challenge bypass token on the user's computer. A computer will have a token if the user previously answered a challenge question successfully from that computer. To SiteKey, the token means "I have seen this computer and user previously, and the user correctly answered one challenge question."

   If a token is found, SiteKey skips the Challenge Q&A (skip to Step 5c).
3. SiteKey poses one of the user's three challenge questions (e.g. "What was the name of your first pet?").
4. The customer answers the challenge question.

5a. If the answer is incorrect, SiteKey asks again. The account is locked if too many wrong answers are entered.

5b. If the answer is correct, a bypass token is generated and saved to the computer as (1) a cookie in the browser; and (2) in the Flash persistent store located on the user's hard drive.

5c. The customer is shown the SiteKey image and text phrase, which he or she confirms is correct.

6. The customer enters the password and clicks "Sign In." The account is opened for access.

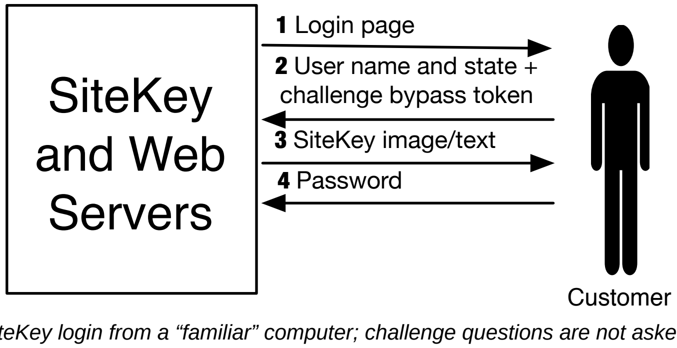


Fig. 5: Alternative SiteKey login from a "familiar" computer; challenge questions are not asked if a token is presented

Technical notes about selected steps in the SiteKey process (Fig 4,5), cookies, and Flash storage

1. Server sets general session cookies in the browser:

```
BOA_0020=(deleted); expires=Sun, 27-Sep-2037 00:00:00 GMT; path=/; domain=.bankofamerica.com;
CFID=(deleted); expires=Sun, 27-Sep-2037 00:00:00 GMT; path=/;
CFTOKEN=(deleted); expires=Sun, 27-Sep-2037 00:00:00 GMT; path=/;
GEOSERVER=1; path=/;
```

2a. Submit button sends form data:

```
reason          (empty)         pswd            (empty)         sitekeySignon   true
Access_ID       (user name)     from            homepage        dltoken         (empty)
Access_ID_1     (empty)         Customer_Type   MODEL           id              *******
Current_Passcode (empty)        pmbutton        true            state           MA
acct            (empty)         pmloginid       pmloginid       country         null
```

2b. Server sets login-related cookies in browser:

```
state=MA;Expires=Wed, 09-May-2007 16:26:37 GMT;Path=/
GSLSESSIONID=(deleted);Path=/
```

2c. ... then runs a small Flash script and checks cookies to look for a valid bypass token:

```
Cookie: PMData="PMV2AAhOQWNc(truncated);Expires=Wed, 09-May-2007 20:55:50 GMT;Path=/"
Flash shared object: PassMark="PMV2AAhOQWNc(truncated)"
```

4. Submit button sends form data:

```
nextAction                    sitekey_confirm
sitekeyChallengeAnswer        (deleted)
sitekeyDeviceBind             false
```

5a. If the challenge was correctly answered, the server stores a bypass token so questions won't be asked again:

```
Cookie: PMData="PMV2AAhOQWNc(truncated);Expires=Wed, 09-May-2007 20:55:50 GMT;Path=/"
Flash shared object: PassMark="PMV2AAhOQWNc(truncated)"
```

5b. ... and displays the SiteKey image and text. The SiteKey image is retrieved via a GET with two encrypted parameters, "it" and "iv". The companion text phrase is sent as inline plaintext without protection:

```
Your SiteKey Image Title:
<TD width="70%" class="text1" style="font-weight:bold">&nbsp;<br>Ready Freddie</TD>
...
<img src="getMySiteKey?it=CF9E4DDF8B5AD60BC64A38A(truncated)&iv=2A9(truncated)
               alt="Travel & Culture 10538">
```

Notes about the challenge-bypass token, its safety in the Flash persistent store, and path obfuscation

Acquisition of a "challenge bypass token" (the cookie "PMData" or the Flash shared object "PassMark,") is the primary goal of an attacker. Given a token and companion user name, an attacker can retrieve and proxy a user's SiteKey image, creating high confidence that a web page is legitimate and the user is safe.

In Flash 8, a domain- and script- specific file such as:

```
~/Library/Preferences/Macromedia/Flash Player/#SharedObjects/ \
(random)/bankofamerica.com/sas/sas-docs/html/pmfso.swf/PassMark.sol[15]
```

holds the SiteKey token. "~" is the user's home directory, and "*(random)*" is a random string such as "M22L7FRG" intended to make the persistent store difficult to find.[16] This adds obscurity, but no security. An attacker with filesystem access need only search the #SharedObjects/ directory to find a desired file.

Actual security comes from Flash Player's ban on (1) filesystem access by Internet scripts; and (2) cross-site scripting. The overall safety of the SiteKey token is a function of Flash Player, operating system and browser security, which block access to local files. The token is reasonably safe if these things are working correctly.

[15] As observed on Mac OS X 10.4 using Adobe Flash Player 8.

[16] Macromedia Flash Player 8 Security White Paper, Adrian Ludwig, Sept. 2005, p.16
http://www.adobe.com/devnet/flashplayer/articles/livepage.apple.comflash_player_8_security.pdf

Vulnerabilities in SiteKey

The primary purpose of SiteKey is to protect against phishing. Many, perhaps most, phishing sites are naïvely *helped* by the legitimate sites they are faking, as follows: the fraudulent server sends out *only* an altered HTML page, with a few changes to divert a victim's secret information to the phisher. The page's other images, links, and multimedia are sent to the victim directly from the legitimate server (Fig. 6). Attackers thus do not need fast servers or fast Internet connections to service thousands of victims. The targeted vendor provides those when its servers cannot distinguish between legitimate and phishing-related requests.

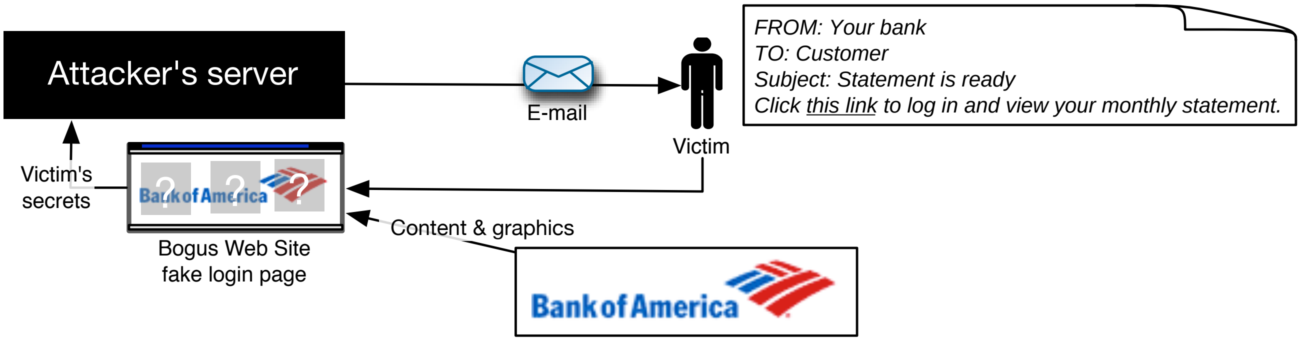


Fig. 6: Phishing e-mail leading customer to fake web site, with pages populated by the legitimate web server

On consideration, SiteKey seems vulnerable to the frauds it is supposed to protect against. Anything a customer's personal computer can do, an intermediary computer (the "man in the middle") could also do. An intermediary attacker could relay a victim's secrets to the legitimate site, pass the output (such as a SiteKey image) back to the victim, and capture the victim's secrets and the site's authentication tokens as they flow past.

News.com reported one response to concerns about such an attack, in May 2005:

> *"Mark Goines, chief marketer for PassMark Security, said that a man-in-the-middle attack is not possible. SiteKey uses a 'secure cookie' to link a user's PC to the Bank of America Web site. The cookie can only be read by a server with a specific security certificate and not by a malicious Web site set up by an attacker in such an attack, Goines said."*[17]

We observed that man-in-the-middle attacks seemed plausible, as did Doug Ross, who wrote in June 2005:

> *"The more I read about Bank of America's solution to the phishing problem, the more I believe it susceptible to man-in-the-middle (MIM) attacks… As far as I can tell, there's no way for SiteKey to distinguish a malicious, zombie PC from a user's virgin computer. The zombie PC could ... proxy login information from the user to the bank and any resulting pages and images from the bank to the victim. Sounds like it's completely susceptible to a man-in-the-middle: the classic phisher's false store-front."* [18]

A June, 2006 summary of the state of SiteKey released by the anti-phishing vendor phishcops.com said:

> *"With significant increases in customer support calls reported, widespread customer resistance, and hints of ongoing implementation and maintenance problems, it appears that PassMark SiteKey may be creating more headaches for Bank of America than it has solved… Although SiteKey has been in operation for over a year, Baseline magazine asserts 'attempted phishing attacks have not decreased' against the bank. One fraud tracking organization the magazine contacted reported no less than 350 attempted phishing attacks launched against the bank since December of 2004."* [19][20]

The WikiPedia entry for Bank of America's Sitekey says in part:

> *"The use of an image is already being hacked... (by) screen scraping on a regular basis."* [21]

[17] Bank of America's SiteKey scrutinized, News.com, May 27, 2005, http://news.com.com/2061-10789_3-5723556.html

[18] Making Phishers Solve the Captcha Problem, Doug Ross@Journal blog, Jun 1, 2005 http://directorblue.blogspot.com/2005/06/making-phishers-solve-captcha-problem.html

[19] Bank of America and Passmark SiteKey: Trouble in Paradise? Jun 8, 2006, http://www.prweb.com/releases/2006/6/prweb394782.htm

[20] Bank of America Seeks Anti-Fraud Anodyne, Baseline Magazine, May 15, 2006 http://www.baselinemag.com/print_article2/0,1217,a=178262,00.asp

[21] http://en.wikipedia.org/wiki/Sitekey#SiteKey, as of July 17, 2006

Finding 1: SiteKey authentication can be circumvented

SiteKey as deployed by Bank of America, and probably as deployed by others, does not provide much outward protection from typical phishing scams. It is true that a scammer cannot guess the secret image and phrase, but these can be easily acquired by an intermediary who relays web pages, cookies and web forms between a victim and the SiteKey server.

SiteKey is susceptible to a real-time, man-in-the-middle attack. An attacker can create a fake web site that looks like a legitimate Bank of America web site, including a victim's correct SiteKey secret image and text phrase.

Therefore, the presence of a SiteKey image on a web page does not conclusively prove that the page is legitimate. Customers of web sites using SiteKey authentication are not invulnerable to phishing frauds enabled by man-in-the-middle attacks.

SiteKey-enabled sites are somewhat safer than sites that don't use the product, because they are harder to fake. In the presence of SiteKey, an attacker must create a "proxy" to alter the pages slightly, and to relay some information between a victim at the fake site, and a real BofA web server. A server- or bot-based script to carry this out would be small, fairly simple, and well within the means of any attacker seeking a financial payoff.

Complicating matters, BofA's instructions to customers could help criminals by persuading skeptical users that a fake site is legitimate. The language is at least inaccurate, and at worst, dangerous, as it may create an expectation of perfect security:

> *"If you recognize your SiteKey, you'll know for sure that you are at the valid Bank of America site. Confirming your SiteKey is also how you'll know that it's safe to enter your Passcode and click the Sign In button."* [22]

How it's done: displaying a valid SiteKey image and text phrase on a third party web page

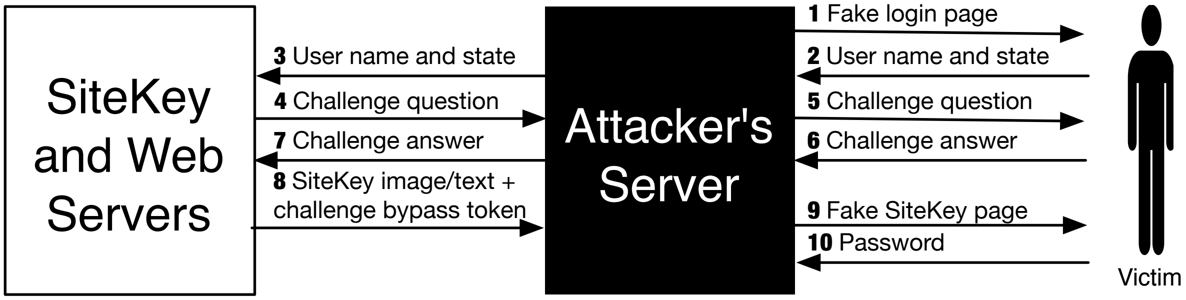


Fig. 7: Proxied SiteKey login

0: Attacker creates a fake web page on an attacker-controlled server.
1: Victim is lured by an e-mail, and loads the fake page.
2: Victim enters user name and state, as if at the real site.
3: Attacker "logs in" to the legitimate server, relaying the user name and state.
4: ... and receives one of the victim's challenge questions.
5: Attacker presents the challenge question to the victim.
6: Victim enters the answer to the challenge question, as if at the real site.
7: Attacker relays the challenge answer to the legitimate server.
8: .... and receives the SiteKey image, text phrase, and challenge-bypass token.
9: Attacker presents the fake "sign on" page to the victim, complete with the correct SiteKey image and phrase.
10: Victim enters the account's password.

The attacker now has everything needed to access the victim's account at will.

[22] Bank of America's SiteKey sign-on page, https://sitekey.bankofamerica.com/sas/signonSetup.do , as of July 17, 2006

Creating a man-in-the-middle attack server for SiteKey

The attacker's server must accept and return cookies to and from the true servers, as a browser would. This allows it to talk unchallenged to the SiteKey server "as if" it were the victim it is impersonating.

When the server retrieves pages from the legitimate web and SiteKey servers, it must rewrite the "Submit" action on all pages that send forms, so the form data goes to the attacker, not to the bank.

If it is like many other phishing servers, the middle-man server will rewrite URLs for images and other content (e.g. <img src="images/picture.jpg">) to fully-qualified form (e.g. <img src="http://bankofamerica.com/images/picture.jpg">), so that the legitimate server sends images, multimedia, and other content that support the fakery.

In short, the attacking server is little more than a proxy that selectively rewrites some key URLs on the pages it handles, and records some of the passing data. To the legitimate web and SiteKey servers, it appears to be a web browser operated by the victim. To the victim, it appears to be the bank's web and SiteKey servers.

Ease of implementation of exploits

The author built from scratch a simple proof of concept proxy server that implements a man in the middle attack as documented here. The design is not sophisticated, and is best described as a simple web proxy that uses a handful of regular expressions to rewrite selected HTML tags on the proxied pages. The proof of concept was easily implemented in the PHP 5[23] scripting language with the CURL "Client URL" library[24].

Differences between a legitimate SiteKey page and a fake, proxied SiteKey page

A proxied SiteKey web page shows only small clues that it's not genuine. The differences are insubstantial, and only one clue differs from what a user would encounter on a faked version of a non-SiteKey web site.

The SiteKey-specific difference is that an attacking site lacks a challenge-bypass token and thus would not be "recognized" as the victim's computer would be, so the victim will have to answer a challenge question. This could raise the suspicions of those who are certain that their computers should be recognized. However, the presentation of a challenge question may not seem unusual to many customers. SiteKey's "remembering" a computer is optional, and some will choose not to use the feature. Others will not have the required Flash Player. Thus, some customers always answer challenge questions at login.

Customers may not fully understand when they should expect to be questioned, and when the appearance of a question is itself cause for concern. Considering the bank's continuous emphasis on security, SiteKey users may be reassured, rather than surprised, to encounter challenge questions from time to time.

Second, the host/domain name in the browser's location bar will be incorrect. But this is true for *all* phishing attacks. In this regard a proxied SiteKey page is no different from an ordinary phishing page. In fact, we suspect that SiteKey users may be more easily deceived: seeing the SiteKey secret image will boost confidence in the legitimacy of a web page, even if it is a fake.

Third, phishing sites do not always use SSL, so the "padlock" icon will not appear. However, because Bank of America and others do not encrypt their home pages, even legitimate home pages do not show the lock icon despite Federal Trade Commission guidance to the contrary.[25] If the bank's home page were SSL-encrypted, users would have no better protection than they would have without SiteKey, but frauds originating at non-SSL servers would be plainly revealed. Bad guys can acquire SSL certificates, or take over a server that already has SSL capability, but plain, non-SSL connections are, like the second item, a common feature of many phishing attacks with or without SiteKey.

[23] http://php.net/

[24] http://us2.php.net/curl/

[25] http://www.ftc.gov/bcp/conline/pubs/alerts/shopalrt.htm

Finding 2: Persistent tokens stored by SiteKey may facilitate attacks and confuse users

The significance of challenge questions in SiteKey

The user-configured challenge questions (such as "What was the name of your first pet?") have been described as a linchpin of SiteKey security, providing protection even if a user's login name and password are stolen.[26,27] However, the user name, password and challenge questions generally "travel together," so that an exploit that can capture two items, can likely capture all three. Sometimes a token is present that bypasses the questions altogether. In that setting, discussed next, only a user name and password are needed to log in; the extra security provided by the (un-asked) challenge questions is irrelevant.

Bypassing the challenge questions, forever, with a token

SiteKey can store a long-lived "bypass token" on a computer that tells SiteKey: "This computer is safe. The user has successfully answered a challenge question from it previously." When a bypass token is present, the challenge questions are skipped, and only a user name and password are required to log in to the accounts to which the token is related. A token can be permanently stored on any computer on which a user has successfully answered a single challenge question.

The tokens reference the server's private list of token/username pairs that aren't asked challenge questions at login. When users share a computer, they also share its token. However, each user's SiteKey question/no-question status is kept separate by the SiteKey server.

Token storage and retrieval

Tokens are stored on the computer as both browser cookies and as "persistent shared objects" in Flash versions 6 and newer.[28] Browser cookies are a guaranteed-available means of storing the token; they offer the simplest means of conveying a token from a web browser back to a SiteKey server. Persistent shared objects are similar to cookies, but work with Flash Player, don't follow exactly the same rules that browsers follow for handling cookies, and unlike cookies, are difficult for users to delete. This makes them desirable as long-lived identifiers, but undesirable in that the SiteKey authorizations connected to them may remain valid much longer than a user intends or understands.

Considerable effort is expended to retain the token. For example, if a token-bearing browser cookie is deleted by a user, it will still exist in the Flash persistent object store. The Flash copy will be found, and the browser cookie re-created, the next time the user logs in through SiteKey. This means that the SiteKey authentication of a computer through one browser (e.g. Firefox) will continue if the user changes to another browser on the same computer (e.g. Apple's Safari). This behavior supports the intuitive explanation that "the computer" (not "the browser") has been "authorized," but may be undesirable due to the silent persistence of the token.

Composition of the bypass token

The bypass token is not cryptographically secured. It is simply a large random number.[29] Even if the token were created by secure means, that would not improve security, as its *usage* in SiteKey is not secure. The token cannot be protected from a man in the middle attack when used with ordinary web browsers, cookies, and Flash. It also is not protected against theft. Transfer of a bypass token from one computer to another renders the second computer capable of signing on to SiteKey without challenge questions for all users related to that token, even if the user whose token is transferred has never used the second computer.

Examination of a handful of bypass tokens suggests at least 600 bits of randomness in the encoded 7-bit values. A brute force attack is unlikely to succeed against SiteKey because an attacker would have to scan the entire random number space for every user name of interest, an infeasible task. The primary vulnerability concern with the bypass tokens is their randomness, which is not evaluated in this paper.

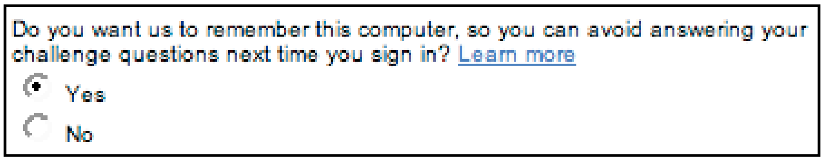
[26] SiteKey FAQ: "What if someone steals my Passcode? How will SiteKey prevent them from accessing my account?" http://www.bankofamerica.com/onlinebanking/index.cfm?template=site_key#passcode

[27] Increasing the number of items required to log in decreases the odds of a successful intrusion to an account.

[28] SiteKey FAQ: "Does SiteKey Use Flash Objects?" http://www.bankofamerica.com/onlinebanking/index.cfm?template=site_key#sitekeyflash

[29] SiteKey FAQ: "Does SiteKey use cookies?" http://www.bankofamerica.com/onlinebanking/index.cfm?template=site_key#sitekeycookies

Security concerns: balancing security and user comfort



Do you want us to remember this computer, so you can avoid answering your challenge questions next time you sign in? Learn more

● Yes

○ No

Fig. 8: SiteKey suggests "Yes" to permanently authenticate a computer for a user name

SiteKey defaults to "Yes, remember this computer" when presenting a challenge question to a user. It is likely that most users will leave a trail of challenge-bypassing tokens on most of the computers they encounter, whether they intended to or not.

This raises serious security concerns for users, and exposes a dilemma for the SiteKey process designer to which there is no entirely adequate solution.

To avoid annoying those users who always sign on from one or a few machines, it is best to remove barriers. People also occasionally forget the answers to their challenge questions, creating work for customer service and frustration for customers who are challenged at inopportune times. Avoidance of the challenge questions is thus desirable and helpful to users and to the bank. Helping users in this way requires a "Yes, remember this computer" preference at the time the token is created.

On the other hand, a bias toward "remembering" a computer indefinitely means that a customer who uses public computer terminals in airports, universities or conferences, or who borrows a laptop intending to make a one-time access to his or her accounts, may inadvertently leave the door to those accounts half-open on machines all across the Internet, with no means of understanding how widely dispersed the tokens are, or of revoking them.

Indefinite replay

Necessarily, tokens are reused for a long time. They can also be transferred from machine to machine. This is an unavoidable and necessary "feature" of the SiteKey design that raises further security concerns. If an attacker is able to get control of *any* token ever generated on any "recognized" machine for a user, the attacker will be able to bypass that user's challenge questions, from any computer, indefinitely.

Tokens do not require secure connections

Persistent values stored with Flash 8 can have an optional "secure" qualifier, as browser cookies may, restricting them to transfer via the HTTPS protocol to SSL-equipped web servers. This attribute is not set on the SiteKey token in the Flash persistent store.

No easy means of de-authorizing computers

SiteKey has no easy means of "de-authorizing" a computer once an access token has been stored on it, creating a permanent, invisible vulnerability on every computer from which a user accesses a SiteKey secured account. The existence of shared objects, much less the process of deleting them, is not readily apparent to end-users, and imposes an additional burden on customers to handle another housekeeping chore.[30]

No means is provided for customers to examine the set of authorized machines, or to revoke the challenge bypass on a computer that a customer is currently signed on to. Ideally, a user could examine or prune a *meaningful* list of token-bearing machines left in the wake of past logins, but this is a practical impossibility due to the system's design, and because most users would have great difficulty "recognizing" machines they have used in the past based given only an IP address or date/time of last use.

[30] Flash Player TechNote: How to manage and disable Local Shared Objects, Adobe Systems http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=52697ee8#change

Discussion: Why server-side-only authentication cannot save the day

SiteKey cannot save users who are misled by a convincing web page that "seems" safe. Customers are on their own when a connected site is malicious, even when they know what they *intended* to connect to.

The frustration of designing any security system is that bad guys ignore it. A bank, no matter how well-meaning, has little ability to intervene in a direct interaction between a customer and a third party. Thus, solutions that rely on the bank doing all the work, or the bank's servers detecting a fraud, are limited from the start.

A big part of the problem is that SSL certificates conflate identity and security. Wrapped up in the SSL certificate behind every HTTPS connection is the identity of the server or organization that sent the page, as well as the cryptographic keys that secure the connection. Every SSL server in the world – millions of machines – has an SSL certificate holding an identity. But individual users only need to routinely verify the identities of a handful of sites, maybe a few dozen, that hold at-risk information. End users don't usually care what the SSL certificate for Yahoo Groups says, but they do care that they're talking to a real bank. Worse, end users have no idea what a valid SSL certificate *should* say, and lack both the skills to interpret them, and the time to bother. Customers don't "authenticate" a local bank branch on the street. Online commerce needs to rise to that same level of implicit, invisible, automatic safety.

Confusing designs further complicate online security. We have talked so far about fake pages that look real, but real pages that look fake also confound customers and interfere with good security. Examples of real web pages that seem fake include the BofA home page that does not use SSL, and Citibank's credit card home page at the distinctly non-Citibank address *https://accountonline.com*.

SiteKey raises the bar a bit, forcing a fraudulent server to engage in a brief interaction with a real server and forcing scammers to write software. It is an example of a security mechanism that tries to protect both ends of a connection by placing all the security at one end. The approach is superficially appealing: if it worked, vendors could deploy server-side security and wipe out fraud without having to educate or support their customers.

One sided security can foil some attacks or raise their costs,[31] but one sided authentication, even if perfectly executed, cannot protect against many attacks, because web browsers are not televisions that faithfully show whatever is sent to them. Rather, browsers are *display engines* that interpret web page codes. They are affected by local settings and operating environments that vary from machine to machine. Even a simple web page may include JavaScript, Cascading Style Sheets (CSS), and plug-ins such as Flash, all of which are interpreted by the browser to turn the page that's *sent* into the page that's *displayed*, probably a little differently on every computer.

Independent connections between a client and server cannot authenticate one another without external coordination. The retrieval of an HTML page (such as a SiteKey page) followed by the retrieval of content for that page (such as a SiteKey image), are independent events. Because of that, an image cannot authenticate its surrounding page without assistance.

The translation from digital (transmitting bits) to analog (reading a screen with eyes) is imprecise. Not only are computing capabilities different from computer to computer, but comprehension and attentiveness vary from person to person and may be compromised by mood, surroundings, and other factors. Successful frauds exploit human limitations with tricks such as the homograph attack, which creates familiar-looking domain names out of international characters. To a computer, the names are very different. To a human, they are indistinguishable.[32,33,34]

A final consideration with SiteKey is that it may not scale comfortably. If just one bank uses it, a customer can memorize one image, one text phrase, and three questions and answers. But what is the online experience like for someone who interacts with a dozen SiteKey-protected sites? For safety, the secrets and answers must be different for each one. Could anyone keep that many details sorted without frustration?

[31] "Cost" may mean financial cost, or complexity - in computation or design of an exploit.

[32] Experts: international domain names may pose threat, Paul Roberts, IDG News Service, in PC World, 08/02/2005 http://www.pcworld.idg.com.au/index.php/id;1052848683;fp;2;fpid;1

[33] "The Homograph Attack", Communications of the ACM, 45(2):128, February 2002 http://www.cs.technion.ac.il/~gabr/papers/homograph.html

[34] http://www.shmoo.com/idn/

Lessons from SiteKey

SiteKey's use in a large installation shows that customers can take an active role in self-protection. This contradicts conventional wisdom about online services, but should not come as a surprise. In retail, cashiers count the money handed to them, and customers count the change given back. We balance checkbooks and dispute errors on credit card statements. Given tools, information, and means of putting those devices to work, customers can and will actively participate in their own protection. However, this requires more effort than looking at an address in a toolbar, and more information than SiteKey provides.

What's missing is the empowerment of end users with appropriate and provably secure methods to detect and respond to frauds in progress. Customers are not stupid. They are human, however, and humans have built-in limits of attention and ability that cause them to occasionally make mistakes or fall for tricks. These limitations must be considered in the design of any security system that places humans in the loop.

A fundamental problem with many online security systems is that they work perfectly when nothing is wrong (when they are not needed), and imperfectly at other times. SiteKey raises concerns about promoting high confidence in security methods that cannot always provide the advertised protections exactly as described. False negatives – when a system fails to detect frauds in progress – increase the risk that overconfident users will be misled by criminals. False positives – when legitimate users are locked out due to innocent mistakes – annoy customers and undermine their confidence.

Suggestions

Single-ended authentication methods invite an arms race mentality with customers caught in the crossfire between criminals and web sites. While strong authentication with active testing at both ends of a connection could cripple phishing and improve the safety of e-commerce, the approach is not yet in favor. The suggestions below may help mitigate some immediate concerns about SiteKey.

Strategies to improve users' understanding of SiteKey's limitations and reduce security risks

1. Avoid marketing claims and instructions that suggest that the appearance of a SiteKey image absolutely proves that a web page is legitimate. As this paper explains, the statement "If you recognize your SiteKey, you'll know for sure that you are at the valid ... site" is not necessarily true all the time.

2. If fine-grained token management is impractical, let customers periodically tell SiteKey to "forget" all the computers they have ever authorized with challenge-bypass tokens. At the least, customers should be able to set an expiration timer so that challenge-bypass tokens periodically expire.

3. Following FTC guidance, serve all pages – including home pages – over SSL. Add hardware accelerators to servers if necessary to handle the load. Customers can then be taught that every page is an SSL page, and to "look for the lock." Some criminals control SSL servers, but most do not. In the future, most will not. SSL isn't a cure-all, but it makes it difficult to use bot nets or to quickly move a bogus site from server to server.

4. Do not store the persistent challenge-bypass token until the user has logged in completely, including entry of a correct password. This does not help with man-in-the-middle attacks, but it does defer the permanent registration of a computer until the person at the keyboard has been fully authenticated.

5. When a user is exiting online banking after connecting from a new computer, confirm their desire to retain a challenge-bypass token on that machine. Follow "Sign off" with "Should we remember this computer ... ?".

6. Limit the number of bypass tokens that can be active for a single account. Make the transfer of a token from one computer to another a big deal, on par with "authorization" of a computer for Apple iTunes (where the stakes are much lower).

7. Enable the "secure" flag on all cookies and set the "secure" option on persistent objects in Flash 8 and later versions (the "secure" option is new in Flash 8).

8. Provide security testing tools (software and/or hardware) that customers can use to obtain real assurances that things are safe. The author and others are building anti-fraud systems that support end-to-end cryptographic authentication across a connection, closing the loop and making successful attacks substantially more expensive to carry out, and less likely to succeed.

End notes

Frequently asked questions

Q: What if HTTPS were always used to show the SiteKey image, the URL for the SiteKey image didn't immediately reveal which image it was, and SiteKey started sounding alarms whenever it got a request for a SiteKey image with a clearly illegitimate HTTP "Referer" header? (Not all clients send "Referer," and not in all contexts, but many do most or all of the time.) This would defeat many more phishing attempts than not checking "Referer" would – the man-in-the-middle would have to proxy everything, incurring heavier load. It might also help BofA quickly identify which hosts are bad. Or maybe they already check "Referer"?

A: Checking the "Referer" header might stop some unsophisticated attacks, and make creation of exploits harder. But a careful man-in-the-middle will transmit valid Referer headers. Also, because some clients don't send Referer headers, or get them wrong, this could inconvenience customers (many people) more than bad guys (a few people). The image URL is encrypted and does not reveal which image is sent out. However, the "alt" text (provided for vision-impaired customers) does expose a permanent plaintext tag for the image.

Q: Surely BofA can pretty quickly identify evil proxy servers, since multiple users will be logging in rapidly from the same IP address... at least except for evil hosts behind large ISPs with transparent HTTP proxies or NATing.

A: BofA says it's actively monitoring for this sort of thing. But in phishing attacks, most people are not fooled most of the time, so the load from a malicious proxy such as those described in this paper may never be high enough to raise an alarm. Also, some legitimate users connect from networks that obscure the origin of a connection. If suspected proxies were found and blocked: (1) malicious web pages would be moved to unblocked addresses on any of the thousands of "zombie" PCs on the Internet; (2) legitimate customers could be cut off. Also, if connections originate at AOL, all bets are off: the IP address from an AOL proxy can change frequently, so it's an ideal starting point for this sort of attack.

The problem for any organization with thousands or millions of online users is that people move around a lot, and use broken browsers and old gear and odd configurations. "Obvious" solutions can't be employed without eventually impacting just about every customer, and such solutions do not assuredly stop criminals.

Q: Shouldn't BofA tell customers to avoid using "public" computers for online banking?

A: Public computers may be risky for many reasons, but a bank can't tell people not to use online banking when and how they choose to, so this is a non-starter. Some people use public terminals as a matter of course, such as a friend who lives four months a year in Mexico and only recently installed Internet service in his condo there. The underlying problem affects private computers too: if I access my account through a friend's laptop, I may leave behind a permanent token that makes my account less secure than it should be. The portability of tokens, and the trail of security-defeating tokens left behind on casually-used computers, are addressable concerns.

Q: Regarding the comment that customers already participate in their own protection, in the US, many cashiers count the cash handed them, and count out customer's change (at least the bills). Some people pay attention to that, and some don't. I bet that those customers who often count it themselves are in the minority.

A: This was just an illustration drawn from experiences on both sides of the retail counter, to point out that to a noticeable degree, people do keep watch over their affairs when they are able to do so.

About the author

Jim Youll is CTO of Challenge/Response, LLC, a Cambridge, MA startup creating tools for fraud prevention, online security and new forms of electronic commerce. He holds an undergraduate degree in Computer Science, and a Master's degree from the Media Lab at the Massachusetts Institute of Technology, where he studied e-commerce, distributed self-regulating market systems, location technologies and wireless. He is also founder of the non-profit Voting Transparency Project, and a fellow of the Stanford Center for Internet and Society.

Acknowledgments

Thanks to all the reviewers for their efforts in fact checking, reading drafts, and suggesting improvements. Thanks also to Bank of America, RSA Security, and the Stanford Center for Internet and Society.