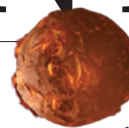


# COMMENT

**BIOTECHNOLOGY** A call for more rigorous research into health impact of GM foods **p.327**

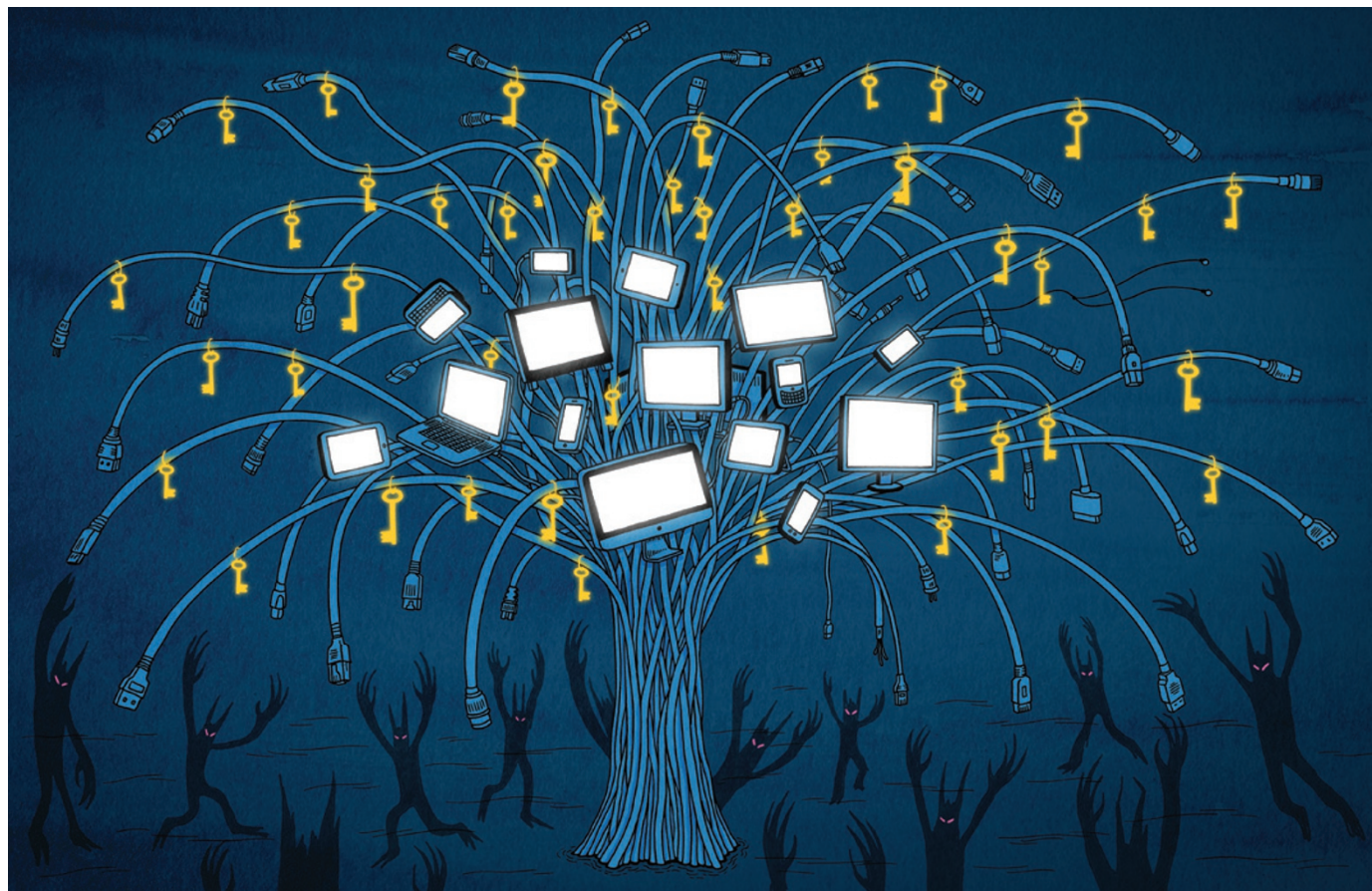
**CULTURE** Conservation in Italy 40 years on from UNESCO heritage list **p.328**

**ASTRONOMY** Extrasolar planets, sci-fi and Kim Stanley Robinson **p.330**



**OBITUARY** Edward Donnall Thomas, bone-marrow pioneer, remembered **p.334**

ILLUSTRATION BY ANDREW RAE



## Secure the Internet

Software engineers must close the loophole used to intercept online communications, say **Ben Laurie** and **Cory Doctorow**.

In 2011, a fake Adobe Flash updater was discovered on the Internet. To any user it looked authentic. The software's cryptographic certificates, which securely verify the authenticity and integrity of Internet connections, bore an authorized signature. Internet users who thought they were applying a legitimate patch unwittingly turned their computers into spies. An unknown master had access to all of their data.

The keys used to sign the certificates had been stolen from a 'certificate authority' (CA), a trusted body (in this case, the

Malaysian Agricultural Research and Development Institute) whose encrypted signature on a website or piece of software tells a browser program that the destination is bona fide. Until the breach was found and the certificate revoked, the keys could be used to impersonate virtually any site on the Internet.

Fake certificates are used by hackers and governments to harvest online communications. In 2011, for example, a hacker based in Iran stole the signing keys from DigiNotar, a Dutch CA that declared

bankruptcy soon afterwards. The keys were used to impersonate sites such as Facebook and Gmail in Iranian dissidents' browsers, allowing all of their messages to be read.

Certificates allow the web to work. They secure transactions and allow users to enter credit-card numbers, share data across networks or chat in private forums. Without certificates, hackers could easily stop, corrupt or eavesdrop on these exchanges. But certificates are in trouble. As more authorizing bodies are added to browsers' lists of trusted CAs, and as governments, ►

► hackers and unscrupulous insiders weaken the Internet's security system, it is becoming virtually impossible to know whether a connection is legitimate.

Many Internet technicians, ourselves included, agree that it is time to fix the problem. What remains is the substantial hurdle of reaching consensus about how.

The international Internet Engineering Task Force is adding registration information to the (already overburdened) domain name system (DNS). But websites can still be taken over. The Electronic Frontier Foundation (where C.D. is a fellow), based in San Francisco, California, has proposed a cryptographic protocol called Sovereign Keys (SK) that would make it impossible for a third party to impersonate any website.

*"It is becoming virtually impossible to know whether a connection is legitimate."*

A third effort is under way, led by a team (including B.L.) at Google, based in Mountain View, California. This protocol — called Certificate Transparency (CT) — is similar to SK, but it includes an independent cross-checking system. Release dates have not been set for either protocol, but CT has the potential to be rolled out sooner, through regular software updates for Google's web browser, Chrome. We see it as a stepping stone to a more ambitious system, such as SK.

We call on browser vendors to support a shift to a more secure system. There are economic barriers — no one is likely to make money from shoring up the Internet. But the risks of ignoring this security loophole are too great.

## CROSS-TALK

Before your browser connects to a website, it asks your local network's DNS server for the numeric address corresponding to the website's domain name. (For example, one of the addresses for [www.facebook.com](http://www.facebook.com) is 66.220.149.11.) But DNS is not secure — its communications with browsers are unscrambled, and they are easy to intercept. Anyone sharing your network can steal your credit-card information or passwords.

To scramble messages and keep them private, you need encryption. If a browser has received a cryptographic certificate signed by a CA, its address bar shows a key or padlock icon. All browsers have pre-installed lists of trusted CAs against which to check certificates.

CAs do a lot of due diligence before issuing a certificate, but they are fallible. Cryptographic methods can be used to spot forgery and tampering, but not to distinguish real certificates issued by diligent CAs from those issued by mistake or by a

CA that has been conned or taken over.

The proliferation of CAs is putting the entire Internet at risk. Governments are not responding to the problem — indeed, some policy-makers have shown a remarkable willingness to undermine online security for law-enforcement reasons. India's government, for example, is seeking weaker security for Skype and BlackBerry mobile devices<sup>1</sup>. In 2011, US lawmakers proposed the Stop Online Piracy Act, which would require DNS providers to return false results when users try to connect to sites accused of facilitating copyright infringement<sup>2</sup>.

Luckily, software engineers are in a position to fix the certificate loophole.

## CERTIFICATE TRANSPARENCY

CT and SK rely on a type of record that uses cryptographic methods to prove that none of its past entries has been erased — an 'untrusted, verifiable, append-only log'. This log is based on a mathematical principle called a Merkle tree — a hierarchy of linked items, or 'leaves'.

In CT, each leaf is a certificate. For a particular node in the tree, a value can be generated in a few steps on the basis of the values of other nodes. For a tree with, say, a million leaves, a verifier would have to track only 20 nodes to confirm any particular leaf. Even a tiny alteration throws off the calculations entirely. With Merkle trees it is possible to prove efficiently that a particular leaf is in the tree without revealing the contents of the other leaves. It is also impossible to fake such a proof for a leaf that is not in the tree.

Merkle-tree logs are stored on a small number of computers, or log servers. Every time a CA generates a new certificate, it sends a copy to all the log servers, which then return a cryptographically signed proof that the certificate has been added to the log. Browsers could be pre-configured with a list of verified log servers (in addition to the list of CAs now installed).

Periodically — perhaps hourly — a number of 'monitor' servers contact the log servers and ask for a list of all the new certificates for which they have issued proofs. These monitors — operated by companies, banks, individuals and service providers — would discover any unauthorized certificates, just as credit reports alert people to cards or loans issued falsely in their names.

This process works only if the log servers are honest; here, auditor servers come in. Every so often, a browser sends all the proofs it has received to a number of auditors — anyone may act as one, because the logs are public. If a proof has been signed by a log server but does not appear in its log, the auditor knows that something is wrong. Within an hour of committing their first transgressions, rogue CAs and

log servers could be detected and removed from browsers' lists.

Even with modest uptake, CT will begin cleaning up the Internet immediately on roll-out. Blocking will improve as more organizations, browser vendors and users participate. Initially, browsers that adopt CT will not be able to block connections for which no proofs are offered. After a year, Chrome will be updated to warn users before establishing a secure connection without a proof. Later, it will not connect to any site without one. We hope that other browsers will follow a similar path.

Also based on Merkle-tree logs, SK is a more theoretical approach. Instead of requiring CAs and domain registrars, SK issues a private key for each website to only one holder. No one else may use that key, so misuse by a third party is not a problem. Anything unverified is blocked, so SK would foil attempts by governments to use domain seizures to censor content that they find to be objectionable.

## LONG-TERM GAIN

Through systems such as CT and SK, software engineers can and should solve the certificate problem. After all, the Internet is international and independent; governments cannot mandate a solution.

There are some economic barriers to improving security, but it is a worthwhile investment. CAs have no short-term incentive to support these measures, but the CA network offers the best avenue for rolling out steps such as CT. Similarly, someone must run the log servers, even though doing so will not lead to direct economic gain. But the long-term stability and security of the Internet is good for business. In line with this view, Google will run some log servers — but others should as well, so that we can avoid putting all of our eggs in one basket. Browser vendors should also commit to supporting CT.

History tells us that those who seek new avenues of attack will eventually find them. But this troubling breach must be closed down now. ■

**Ben Laurie** is a visiting industrial fellow at the University of Cambridge, Cambridge CB2 1TN, UK. **Cory Doctorow** is a visiting senior lecturer in the Computing Department at the Open University, Milton Keynes MK7 6BJ, UK.  
e-mail: [ctnaturepaper@links.org](mailto:ctnaturepaper@links.org)

1. Anonymous. India to seek Interpol help to intercept encrypted data from BlackBerry, Gmail, Skype. *The Economic Times* (20 December 2011).
2. US House of Representatives, 112th Congress, 1st sess. bill no. HR 3261 (2011).

**Competing financial interests declared; see [go.nature.com/cdgvuun](http://go.nature.com/cdgvuun).**