May 16th 1994

Making certificates hard to forge

Srinivasan Keshav

AT&T Bell Laboratories
600 Mountain Avenue, Murray Hill, NJ 07974, USA
keshav@research.att.com

Abstract

I describe a method for making cheap and relatively unforgeable certificates of authenticity. The invention is based on Public Key Cryptography. The invention uses the serial number of a currency note as part of a *digital signature*, and the currency note itself is part of the certificate. Thus, to forge the certificate, one needs to forge a currency note, which is hard to do and easy to detect. The scheme is easy to implement and imposes very little burden on the creator of the certificate. It also allows independent verification of the authenticity of the certificate by third parties without reference to the creator of the certificate.

1. Desirable properties in a certificate

In many real-world applications, it is necessary to create certificates of authenticity that have some useful properties. Let the certificate **C** have an actual creator **R** and say that it has been created by **R'**. We would like the following to be true:

1) It can be verified that **R'** is really **R**. For example, if it claimed that the Board of Education has certified that a student has received the High School graduation degree, it would be nice to know that this certificate was indedd issued by the Board, and not by a obliging forger.

2) **R** does not need to be contacted to verify property 1. Otherwise, the creator of the certificate would be inundated with calls.

3) Copying **C** is hard. Otherwise, a valid certificate could be reproduced without violating property 1.

4) The certificate can be created at low cost.

This type of certificate has several potential uses, but one immediate application is in the art world, where one would like the creator of an art work to create a certificate of authenticity that could be sold to subsequent owners along with the art work. Since each work of art can potentially cost thousands of dollars, the ability to verify the authenticity of the art work (even a print) is important.

2. Solution

The solution is based on three ideas. First, public key crytography allows us to create unforgeable digital signatures. Second, currency notes are relatively unforgeable (indeed, several hundred years worth of effort has gone into making this so). Third, a currency note is a readily available source of unique serial numbers. By combining these ideas, I obtain a scheme that satisfies the requirements in Section 1. I now discuss the solution in more detail.

2.1. Public Key Cryptography

In public key crytptography, each user **U** is given two keys, a public key **P(U)** and an private key **V(U)**. The user can encrypt any plaintext **T** with the private key **V(U)** to create the encrypted text **V(U)(T)**. Subsequently, any other user can decrypt this using **U**'s public key, so that **P(U)(V(U)(T))** = **T**. It is property of public key cryptography that even if given the public key, the private key is extremely hard to determine. Thus, as long as a user keeps his private key secret, he has the ability to create unforgeable *digital signatures*. These are simply plantexts that are encrypted with **U**'s private key. Since no one else can create them, but anyone can check their authenticity, they are the equivalent of **U**'s signature.

2.2. Serial Numbers

One way to make certificates that are unique is to include unique serial numbers in the digital signature. Since the forger cannot create a digital signature with a different serial number, the serial number must also be copied. So, the worst that a forger can do is to create multiple copies of the same digital signature. However, if the certificates are widely dispersed, these copies might be simultaneously in circulation without attracting attention. This is potentially a problem.

2.3. Currency notes

A currency note, such as a dollar bill, is a cheap source of unique serial numbers. Thus, if the digital signature incorporate the serial number of a dollar bill, we instantly obtain unique digital signatures as described above. However, dollar bills have another nice property - they are hard to forge. They often are printed on special paper, with magnetic inks, complicated patterns and watermarks especially to prevent duplication. Thus, if the certificate has a dollar bill stapled on, with the serial number of this bill incorporated into the digital signature, the certificate become as hard to forge as a dollar bill.

3. The Scheme

The creator of the certificate selects a currency note, and notes down the serial number **SN**. Then he creates the following string:

<name of creator><date><name of certificate><serial number of currency note>
<serial number of the certificate><description of the certificate>

This is then encrypted with the creator's public key, and printed onto a sheet of paper. The currency note is attached in some fashion to this certificate. One can easily write a small piece of software that is told about the creator's private key, and a set of currency serial numbers, and automatically prints a number of certificates. The certificate also has, in plaintext, the name of the creator.

A verifier would call up the public key cryptography service, and request the public key of the creator. This would then be used to decrypt the digital signature. If the signature cannot be decrypted, then it is forged. If the signature is correctly decrypted, then the verifier simply matches the serial number of the currency note with the serial number in the signature. If this doesnt match, the certificate has been copied. If it does, the verifier can check if the currency note is in fact valid or not. This can be done at most banks. If all the tests pass, there is a very strong likelyhood that the certificate is authentic.

4. Getting public keys

The crux of the scheme is in a creator getting a public key/private key pair, and in the verifier getting the *correct* public key for the creator (by faking the public key, a forger can fool the verifier). Fortunately, there is a company called RSA Inc, which provides software systems that will allow AT&T to provide these services (see Appendix 1). A public key server is also available on the World-Wide-Wed at URL http://martigny.ai.mit.edu/~bal/pks-toplev.html.

5. Claims

I claim that this scheme satisfies the properties required in Section 1. The digital signature verifies the creator of the certificate. Authenticity can be verified by third parties simply by obtaining the public key, which can be done fairly easily. The certificate is hard to forge because currency notes are hard to forge. The certificate is low-cost: there is only the cost of software, and then one dollar per certificate.

6. Appendix

ABOUT RSA LABORATORIES

RSA Laboratories is the research and development division of RSA Data Security, Inc., the company founded by the inventors of the RSA public-key cryptosystem. RSA Laboratories reviews, designs and

implements secure and efficient cryptosystems of all kinds. Its clients include government agencies, telecommunications companies, computer manufacturers, software developers, cable TV broadcasters, interactive video manufacturers, and satellite broadcast companies, among others.

RSA Data Security, Inc.
100 Marine Parkway, Suite 500
Redwood City, CA 94065-1031

Phone: 415/595-8782 Fax: 415/595-1873 Internet: info@rsa.com

BSAFE: RSA's General Purpose Software Developer's Kit for Cryptography

BSAFE, the encryption "engine" used by the developers of products like Lotus Notes, Novell NetWare 4.0 and WordPerfect InForms, is now available in a major new release. BSAFE 2.0 is a portable C toolkit that allows developers to integrate privacy and authentication features into virtually any application. BSAFE 2.0 modules can be used to construct anything from RSA Digital Signatures to complex key exchange, encryption or key negotiation schemes. BSAFE 2.0 provides the programmer with a complete palette of crypto algorithms, including RSA, DES, the exportable RC2 and RC4 ciphers, Diffie-Hellman, and many more.

Year Introduced: 1986
Current Version: 2.04
Price: $290

Certificate Issuing System

In any RSA-based security system, authenticity of an individual's RSA public key is crucial. The CCITT X.509 Digital Certificate is the internationally accepted method of proving identity and public key ownership across the network, and RSA's Certificate Issuing System provides a complete solution for issuing and tracking digital certificates within your organization. The intuitive CIS Apple Macintosh interface and integrated ORACLE database make CIS easy to install and use in the places in your company that normally verify identity: the personnel departments, badge offices and security desks. CIS complies with X.509, PKCS and Internet Privacy-Enhanced Mail standards.

Year Introduced: 1992
Current Version: 1.1
Price: $14,900