Secure and cost efficient electronic stamps

Detlef Hühnlein and Johannes Merkle

secunet Security Networks AG
Mergenthalerallee 77-81
D-65760 Eschborn, Germany
{huehnlein,merkle}@secunet.de

Abstract. Even small companies do not use physical stamps to prepay postal services, but have a franking machine which can be logically loaded with a certain amount of stamps which are printed on letters for example. Special purpose franking machines, which are still widely used in practice, however have security and handling problems. It was relatively easy to forge stamps and in some cases one needed to bring the franking machine to the post office to get it loaded. Therefore the USPS (US Postal Service) initiated an information based indicia program (IBIP) [1] to provide "electronic stamps". While the two problems mentioned above were solved by applying (asymmetric) digital signatures and a special purpose hardware device on the client side, it seems that this approach introduces an unreasonable overhead. In fact the problem of verifying the huge number of stamps / signatures in reasonable time is not adressed in [1] at all. Therefore many international postal service providers, like the German Post AG for example, are hesitating to implement this concept.
In this short note we will introduce an alternative approach using symmetric algorithms and general purpose smart cards to provide electronic stamps. Thus it will be less expensive to implement this concept. Furthermore our approach will allow an efficient verification of all stamps in a very short time because one does not need to contact a certificate directory. Note that in this special scenario it is no problem at all that only the postal service provider is able to verify the "symmetric signatures" and hence the authenticity of the stamps.

1 Introduction

While emails increasingly replace paper bound mail, there is still a large necessity for conventional postal services and it can not be expected, that the electronic analogue will supersede conventional mail entirely in the future, because emails obviously lack some properties of snail mail. Therefore it is necessary to integrate interfaces to postal services into the existing office communication environment. Even small companies apply franking machines, which are issued by postal service providers (PSP). Currently all such franking machines are expensive special purpose machines, whereby most of them, or at least the security component, have to be carried to a post office to be loaded after prepaying a

certain amount of stamps. The security of this procedure, i.e. that it is not possible to forge stamps, rests in the secrecy of the interfaces and tools. Even more modern franking machines with integrated modems to perform the loading remotely require a secure direct connection to the PSP and some sort of out of band payment for the bought stamps.

It is clear that it would be desireable to use existing office communication components, like multi purpose printers and PCs with connection to the internet to replace the expensive special purpose franking machines and avoid the annoying trip to the post office. But performing the loading process via open networks like the internet and using standard peripherals to produce and print stamps evidently bears some risks. Hence it is necessary to integrate security mechanisms which prevent unauthorized loading of the franking machine and forging or copying of stamps.

Therefore the USPS initiated the information based indicia program [1]. In this concept the authenticity of stamps is ensured by RSA, DSA or ECDSA signatures, which are coded in a two dimensional barcode and printed as part of the stamp on a letter for example. When a letter arrives at the PSP the stamp is scanned and the signature is checked after obtaining the certificate from a directory. Because one has to connect to a possibly remote directory server to look up the certificate to verify the signature this step is certainly the bottleneck in the verification procedure. With current technology it seems impossible to check a non-negligible fraction of the huge amount of letters. Unauthorized stamping is prevented by using special purpose hardware at the client system. Because copying authentic stamps can not be prevented it is necessary to integrate data chararcteristic for the letter like the zip-code, the adress of the receipient and the date into the stamp. Thus, copying stamps only makes sense in circumstances where one needs to send many letters with identical characteristics. The possibility of copying stamps can be further restricted by limiting the time of validity. However, one can still imagine situations where illegal duplication of stamps may be a concern. Therefore it will be neccesary to log the verified and unexpired stamps.

As noted above the application of digital signatures and accompanied public key infrastructures introduces an unreasonable overhead in the verification step. Because the signatures are exclusively checked by the PSP it is clear that one may as well use symmetric algorithms with derived keys to obtain the same security features. This alternative approach, which is discussed in this work, will allow to check all arrived letters during the sorting process. Furthermore we will see that the special purpose hardware device with realtime clock is not necessary. Hence it will be much cheaper to implement our concept compared to [1].

This paper is organized as follows: In Section 2 we will briefly explain the central features of USPS's information based indicia program [1] and point out the deficiencies for broad application. In Section 3 we will introduce our approach using symmetric algorithms and general purpose smart cards.

1.1 Previous work

There have been several publications treating the realization of secure electronic stamps. In [2] Pastor outlines how such a system might work. In [3] Tygar and Yee give a detailed discussion of the requirements and possible solutions, but in contrast to our concept they only consider protection by digital signatures. Furthermore there is a patent application on cryptographically secured electronic franking systems [4].

2 USPS's information based indicia program (IBIP)

In order to facilitate electronic franking and prevent fraud the USPS initiated IBIP [1]. In this program the authenticity of an information based indicia (electronic stamp) is ensured by applying cryptographic mechanisms to data which are related to the piece of mail under consideration. In the following we will briefly highlight the main issues of IBIP and point out the problems for large scale application.

2.1 A brief overview of IBIP

Every customer who is willing to use electronic stamps buys a *Postal Security Device (PSD)* as specified in [1, Part B]. This device is a special piece of cryptographic hardware with real time clock, which can be connected to the parallel port for example. "For security reasons, the PSD will not be a generalized digital signature device." [1, page B-4]. For the private key in the PSD the USPS creates a certificate containing the corresponding public key, which is stored in a certificate directory. The PSD can be loaded with a certain amount of stamps, by connecting to the PSP and triggering some sort of payment mechanism for the stamps. A loaded PSD is then used to issue electronic stamps which are coded in a two dimensional bar code and printed on the letter. The stamp as specified in [1, Part A] consists of 49 bytes letter specific data (e.g. customer ID, date of mailing, destination, postage, serial numbers, ...) and a digital signature of these data which is generated by the PSD using its private key. The allowed signature mechanisms and key sizes are 1024 bit RSA, 1024 bit DSA or 160 bit EC-DSA. Thus the size of the machine readable stamp is 177 bytes for RSA or 89 bytes when using DSA-type signatures. Hence one has the choice between a barcoded stamp of reasonable size (DSA) or efficient verification (RSA). The part of the program which is not yet specified in [1] is the verification procedure at USPS. This verification step will need to consist of reading the barcoded stamp, connecting to the directory and verifying the signature. It is clear that there will be billions of letters which have to be handled by the USPS every day. Thus it is clear that for performance reasons it will not be possible to verify every stamp. This may lead to "calculated" fraud.

2.2 Summarizing IBIP's problems for large scale application

In this section we will briefly summarize the major problems of IBIP for large scale application:

- IBIP requires special purpose hardware which leads to higher initial costs and hence may deter potential customers.
- The barcode which carries the stamp is relatively big which leads to problems when stamping regular letters or postcards.
- When using DSA-type signatures one obtains smaller signatures and barcodes but has to perform a less efficient verification procedure.
- In both cases (RSA, DSA-type) one needs to look up the certificate in a directory, which makes the verification procedure inefficient and makes the verification of all stamps impossible.

3 A more reasonable concept to realize practical and secure electronic stamps

In this section we will introduce a way to realize electronic stamps which solves the above problems without reducing security. In contrary our approach allows much more efficient verification and hence makes the verification of all stamps possible which should lead to even less fraud. In our concept we assume that every customer has access to a PC equiped with a printer, a smart card reader and an internet connection. It is widely believed that within a few years smart card readers will become standard for PCs. Further, the PSP issues to all customers, who want to use electronic stamps a dedicated software, called the estamp program, and a smart card. Each smart card has implemented a symmetric encryption function F_{SK} and has securely stored a distinguished secret key. This secret key is derived from a master key of a local PSP office (e.g. for every ZIP-code) and the customers ID using an arbitrary secure hashfunction h or alternatively a symmetric cipher in hash-mode. Further, each smart card has 2 internal counters z_1 and z_2 , which cannot be accessed from the outside. Like in the approach of USPS our system consist of 3 stages: Charging the smart card, Generating stamps, Verification of the Authenticity of the stamp by the PSP.

3.1 Charging the smart card

Before a customer can create electronic stamps, he has to charge his smart card. This is initiated by sending the command GEN_REQUEST together with the amount x to the smartcard. The smart card increments the internal counter z_2 and using its secret key to compute the encrypted charge request. This request contains the counter z_2, the amount x, the customers ID and the keyword REQUEST. Then the estamp program sends this request to the local PSP office (e.g. via email, http, tcp).

After receiving this request, the PSP derivates the customers secret key from its local office key (which itself may be derived from a global master key) and

decrypts the request, thereby verifying its authenticy. If positive, the PSP uses the customers secret key to generate an encrypted charge command containing the counter z_2, the amount x and the keyword CHARGE and sends it to the customer.

Finally the customer forwards the received message to the card, which decrypts the charge command. If the charge command is verified the card increments its counter z_1 by x.

1. The customer sends (GEN_REQUEST, x) to the card.
2. The card sets $z_2 = z_2 + 1$ and sends $Y := F_{SK_C}(\mathsf{REQUEST}, z_2, x)$ to the customer.
3. The customer sends (Y, ID_C) to the PSP.
4. The PSP computes $SK_C = h(SK_{LO}, ID_C)$ and $(\mathsf{REQUEST}, z_2, x) = F^{-1}_{SK_C}(Y)$.
5. The PSP sends $Z := F_{SK_C}(\mathsf{CHARGE}, z_2, x)$ to the customer.
6. The card verifies $F^{-1}_{SK_C}(Z) = (\mathsf{CHARGE}, z_2, x)$. If this is ok it sets $z_1 = z_1 + x$.

3.2 Generating stamps

When a customer wants to stamp a letter, he uses his estamp program to send a stamp request containing the postage amount y (which can be determined by the estamp program), a hashvalue v of the specific parameters of the letter and the keyword STAMP to the smart card. The specific parameters of the letter contain the address of the receipient, the customers ID, the date and may contain other data as well making the specific parameters unique. Note that the customer is responsible to use the correct date. If the date is not within a specific time frame when checked at the PSP the letter is considered more closely, as it might have a fraudulent stamp. Thus if the date is not correct the letter will take longer time to be delivered. The card checks $z_1 \geq y$ and, if positive, decrements z_1 by y and generates the stamp for the letter, by encrypting v concatenated with y. If $z_1 < y$ the card returns EMPTY.

Finally the stamp and the specific parameters of the letter are printed onto the letter using an arbitrary machine readable encoding.

1. The customer determines the postage amount y and the specific parameters of the letter D, calculates $v := h(D)$ and sends (STAMP, v, y) to the card.
2. The card checks $y \leq z_1$. If positive, it sets $z_1 = z_1 - y$ and sends $X := F_{SK_C}(v, y)$ to the customer. If negative, it sends EMPTY.
3. The customer prints (D, X) onto the letter.

3.3 The verification of the validity of stamps by the PSP

The PSP can verify the validity of stamps without connecting to a database for every stamp. First it checks that the specific parameters are consistent with the letter (e.g. that the address and the date is correct). If the date is not within a certain time frame (i.e. dated in the future or older than (say) three days) the PSP redirects the letter to a place where stamps which might have been forged are considered more closely. Only in this case the PSP stores the suspicious stamps in a database to recognize copied stamps. Note that the customer itself is responsible for the date in the stamp to allow timely processing without second level checking. This strategy makes the presence of a secure real-time clock at the client system obsolete. After this checking the PSP computes the customers secret key using the secret key of the local office and the customers ID contained in the specific parameters of the letter. Using the customers secret key the PSP decrypts the stamp yielding a pair (v, y). Finally, it checks, that amount y is sufficient as postage for this letter and that v is the hashvalue of the specific parameters of the letter.

1. The PSP reads (D, X) and checks the consistence of D (e.g. consistence of address, expiration of validity).
2. The PSP extracts ID_{C} from D and computes $SK_{\mathrm{C}} = h(SK_{\mathrm{LO}}, ID_{\mathrm{C}})$.
3. The PSP computes $(v, y) = F^{-1}_{SK_{\mathrm{C}}}(X)$.
4. The PSP verifies $v = h(D)$ and checks that amount y is sufficient as postage.

3.4 Replay Detection

Although a stamp is tight to a fixed set of characteristics of the letter, one still has to consider replay attacks. It is not unlikely that a company needs to send many letters with the same characteristics within a short time (e.g. the correspondece with one of its dependencies). In this case the company could save a lot of money by illegally copying and reusing stamps.

The only way to detect illegal copying is to log all verified stamps in databases. Since mail is usually verified at a post office located in the same region as the costumer, this can be done in a decentralized way, i.e. the stamps of a certain costumer are logged at the regional post office. The data of mail which is verified by different post offices can be exchanged by network connections.

Furthermore, since stamps are likely to be verified in essentially the same order as they have been generated, the logging can be done very space efficient: For each costumer C let i_C be the greatest number i for that all stamps of costumer C having serial number smaller than i are either expired or have been already verified. Then for each costumer C it is sufficient only to store i_C and the (compressed) list of the serial numbers greater than i of the verified stamps from costumer C.

For concrete estimates for the expected size of the databases we refer to [3].

4 Conclusion

We will conclude this work by briefly comparing our approach to IBIP:

Advantages of IBIP:

- If a secret key SK_{LO} of a local post office is compromised in our approach the PSP has to replace a set of smart cards – all smartcards whose key is derived from SK_{LO}. In IBIP, if a secret CA-key was compromised, one would only need to replace the certificates signed by this key. This is no real threat as SK_{LO} and the secret CA-key are additionally secured by strict organizational means.
- By using the special purpose hardware with real-time clock it would be harder for an attacker to change the time to produce forged stamps.

Advantages of our approach:

- Our approach does not require special purpose hardware but simple smart cards at the client which is much more cost efficient.
- The "signature" in the stamp in our concept is at most 16 byte which is less than half as big as in IBIP using DSA-type signatures, not to talk about RSA. Thus our stamps are no problems even if printed to postcards or small letters. Thus our stamps do not cause problems, even if they are used for postcards or small letters.
- The verification of stamps in our approach is much more efficient than in IBIP, because one does not need to connect to a directory to obtain certificates, but derives the corresponding symmetric key by simple operations. Thus it will be feasible to verify all stamps, recognize forged stamps and hence prevent fraud.

Comparing the arguments for IBIP and our approach we think that our approach is much more suitable to implement a large scale system for electronic franking.

References

1. United States Postal Service: *Performance criteria for information-based indicia and security architecture for IBI postage metering systems*, August 19th 1998, via http://www.usps.com/ibip
2. José Pastor, *CRYPTOPOST (TM): A universal information based franking system for automated mail processing*, Journal of Cryptology 3 (2):137-146, 1991
3. J.D. Tygar and Bennet Yee, *Cryptography: It's not just for electronic mail anymore*, Technical Report CMU-CS-93-107, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1993
4. World Intellectual Property Organization, International Application Under the Patent Cooperation Treaty (PCT) *System and method for retrieving, selecting and printing postage indicia on documents*, International Application Number: WO 97/14117, April 17th 1997