Report

Blind Signatures

for Untraceable Payments

Project Report on research paper -
Blind Signatures for Untraceable Payments
by David Chaum (1983)

Student: Aaryan Raj Saxena
Student Roll No.: 2020004
Course: Applied Cryptography (CSE546)
Monsoon Semester 2022
December 6, 2022

Summary

This report discusses use of blind signatures for constructing a payment system with untraceable digital payments. We do a literature review of David Chaum's paper on "Blind Signatures for Untraceable Payments" in which he present a new kind of cryptography.

Key findings about this cryptosytsem include:

- *Inability of third parties to determine payee, time or amount of payments made by an individual.*
- *Ability of individuals to provide proof of payment, or to determine the identity of the payee under exceptional circumstances.*
- *Ability to stop use of payments media reported stolen.*

It is, in an essence, the automation of how we pay for goods using traditional payment methods like as bank notes and coins. We also tackles some of the issues identified, including a lack of control, privacy, and security in this report.

The information presented in this report has been gathered from secondary sources, and are cited appropriately for the readers.

The report has been prepared for submission as Course Project of the Applied Cryptography Course at IIIT Delhi.

Contents

1. Introduction

The automation of how we make purchases and pay for products and services is already in progress, as seen by the expansion and diversity of online banking services available to customers. On the one hand, knowing the payee, amount, and timing of a payment by a third party might tell them a lot about the payee's whereabouts, relationships, and way of life. On the other hand, the absence of security and regulations in an anonymous payment system like a bank is problematic. As an illustration, think of issues like the absence of payment documentation, media theft, bribery payments in the dark, tax fraud, and black markets.

1.1 Research question

To present a new electronic payment system addressing both conflicting issues of personal privacy and auditability arising from nature and extent of criminal use of payments.

2. Functions

A blind signature technique is a type of cryptographic protocol that involves two parties: a user named Alice who wants to get signatures on her communications and a signer named Bob who has access to his private signing key. Without Bob knowing anything about the message, Alice successfully completes the procedure and receives Bob's signature on message m.

2.1 Functions Description

David Chaum presents blind signatures as a system that includes the features of true two key digital signatures systems combined in a special way with commutative style public key systems [2]. The following three functions make up the blind signature cryptosystem:

- A signing function s' known only to the signer, and the corresponding publicly known inverse s, such that $s(s'(x)) = x$. and s give no clue about s'.
- A computing function c and its inverse c', both know only the provider, such that $c'(s'(c(x))) = s'(x)$, and $c(x)$ and s' give no clue about x.
- A redundancy checking predicate r, that checks for sufficient redundancy to make search for valid signatures impractical.

2.2 Functions Preparation

Blind signature functions can be constructed from many public key signing protocols. We present here Blinded RSA signatures which is based on RSA signing [3].

Traditionally, RSA signature is calculated by raising the message m to the secret exponent d public modulus N. We modify it by including a random value x chosen by

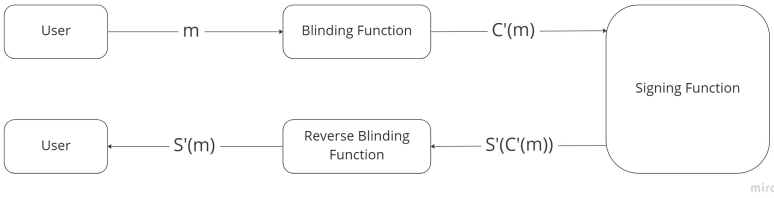

Figure 1: Blind Signature Protocol

sender such that $gcd(x, N) = 1$. The sender of the message 'blinds' the message m by computing $m' \equiv mx^e \ (mod\, N)$ and send m' to signing authority.

Since x is random value and the mapping $x \longmapsto x^e \ (mod\, N)$ is a permutation, it implies that $x^e \ mod\, N$ is also random. This means that m' gives no clue about m. The blinded signature s' is then calculated as $s' \equiv (m')^d \ (mod\, N)$ which is sent to the author (sender) of the message.

To get the unblinded signature s of actual message m, sender removes the blinding factor in the following way:

$$s \equiv s'.x^{-1} \ (mod\, N) \tag{1}$$

2.3 Correctness of protocol

The correctness of the protocol i.e. $s \equiv m^d \, (mod\, N)$ can be obtained from the following equation:

$$s \equiv s'.x^{-1} \ \equiv (m')^d x^{-1} \ \equiv m^d x^{ed} x^{-1} \ \equiv m^d x^1 x^{-1} \ \equiv m^d \ (mod\, N) \tag{2}$$

3. Untraceable Payment System

3.1 Protocol

David Chaum presents the following payment protocol using the functions defined above to create a untraceable payment system:

1. **Initial step by payer** – Payer chooses x at random such that $r(x)$, and forms $c(x)$. Then, the payer forwards $c(x)$ to bank.
2. **Signing step by bank** – Bank signs note $s'(c(x))$, and debits payer's account. Then, bank returns the signed note, $s'(c(x))$ to payer.
3. **Signature verification step by payer** – Payer strips note by forming $c'(s'(c(x))) = s'(x)$. Then, payer checks note by checking that $s(s'(x)) = x$ and stops if false. Now that the payer has a valid note, he can pay anyone with via sending $s'(x)$.
4. **Signature verification step by payee** – Payee checks note by forming $r(s(s'(x))$ and stops if false. Payee then forwards note $s'(x)$ to bank.

5. **Signature verification step by bank** – Bank checks note by forming $r(s(s'(x)))$ and stops if false. Then he, adds note to comprehensive list of cleared notes and stops if note is already on list, credits account of payee, and finally informs payee of acceptance.

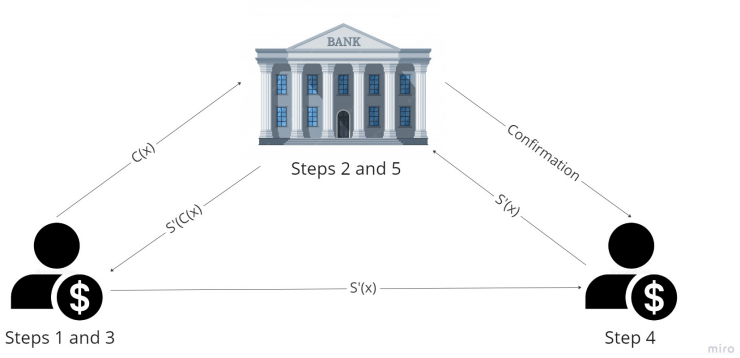

Figure 2: Untraceable Payment Protocol

3.2 Properties

The blind signature system, which combines the aforementioned functions and protocols, is expected to have the following security characteristics:

1. **Digital signature** – anyone can verify that stripped signature $s'(x)$ was formed using signer's private key s'.
2. **Blind signature** – signer knows nothing about the correspondence between elements of the set of stripped signed matter $s'(x_i)$ and the elements of the set of unstripped signed matter $s'(c(x_i))$.
3. **Conservation of signatures** – provider can create at most one stripped signature for each thing signed by signer.

Notice that by virtue of the property of blind signature, when the bank receives a note to be cleared from the payee in step (5) the bank does not know which payer the note was originally issued to in step (2). Moreover the digital signature and related conservation of signatures properties above ensure that counterfeiting is not possible.

3.3 Auditability

The bank's verification process when receiving a note allows the model, as it stands, to identify uncleared notes. By including receipts, we can expand the present model to make it auditable. The payee may even present the payer with a receipt that includes a copy of the note and the specifics of the order. The following is possible as a result:

- By working with the bank, one may determine whether the money was really received by the desired payee or whether there was some sort of fraud involved.
- With the payee's assistance, a note sent to the black market may be tracked to determine the ultimate account it was placed in.

- Using the receipts, one may compute taxes. Verification of spending and the discovery of illicit activity would both be made easy as a result.

3.4 Limitations and Challenges

This blind signature scheme presented above using RSA is prone to RSA blinding attack [1] by which attacker may trick the bank by decrypting a message by blinding another singing message. Note that signing process is equivalent to decrypting with signer's private key. So a message m encrypted with a signer's public key can be provided blindfolded by an attacker for them to sign. The encrypted message m' will often include some sensitive data that the attacker saw being sent under the signer's public key and is interested in learning more about. The attacker will obtain the clear text when they remove the blindness from the signed version.

This attack works because the signer signs the message directly in this blind signing system. Instead of signing the message itself, the signer would typically use a padding scheme in an unblinded signature scheme (for example, by signing the output of a cryptographic hash function applied to the message instead of the message itself). However, since the signer is blinded and does not know the contents of the message, any padding scheme would result in an incorrect value. One solution could be to use different keys for encryption and signing process.

Another solution would be to use other blind signature schemes like Elgamal Blind Signature Scheme (based on the hardness of dLog problem), Schnorr Blind Signature Scheme (based on the hardness of dLog problem), etc.

4. Conclusion

David Chaum really put this idea into action in 1989 under the name Digicash, using eCash as their flagship product. It was successful for a while, acknowledged by Credit Suisse in Switzerland, offered by Deutsche Bank in Germany, Bank Austria, etc. However, it and ecash both filed into bankruptcy in 1998.

In this report we discussed about blind signatures and their application in designing an untraceable payment and voting system. We did review of David Chaum's paper on blind signatures and presented a simple and efficient implementation of the protocol. We also checked the auditability of this protocol. At the end we addressed some of the limitations and challenges associated with this protocol and how can we overcome them.

Bibliography

[1] Rsa blinding attack. URL https://en.wikipedia.org/wiki/Blind_signature.

[2] David Chaum. Blind signatures for untraceable payments, 1998. URL http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF.

[3] S. Goldwasser and M. Bellare. *Lecture Notes on Cryptography*. MIT, 2001.