1. Digital Signatures

1.1 Introduction

A digital signature is "equivalent" of a "paper signature", in the sense that it identifies the individual with a particular document or message. It is different from "paper signature" in the sense that it varies according to the document or message being signed. Since they are message dependent, they also serve the purpose of ensuring the integrity of document or message being sent.

Thus, a digital signature is both signer-dependent and message-dependent.

1.2 Generation of Digital Signatures

Digital signatures are generated using public key cryptographic algorithms. Fig 1 describes a method for generating a digital signature.

Digital signatures are generated using one's secret key. First, the document or message to be signed is hashed to generate a message digest. The message digest is then encrypted using the secret key. The encrypted message digest is the digital signature.

![Image of digital signature generation process]

Fig 1

1.3 Verification of Digital Signatures

Fig 2 describes the process of verifying a digital signature. The receiver, upon receiving the document or message, decrypts the digital signature using the sender's public key to get the message digest. The receiver then hashes the received document or message. The message digests are then compared to see if they match.
1.4 RSA Signature Generation

Digital signatures are generated using RSA algorithm. To generate a signature using RSA, assume that \(<e, n>\) is the signer’s public key and \(<d, e>\) the signer’s secret key. The steps in generating and verifying the signature on message \(m\) are

1. Generate signature \(s, s = m^d \mod n\).
2. Verify signature on \(m\) by using public key \(<n, e>\), by checking \(m = s^e \mod n\) since \(de = 1\).

Note that the RSA signature generation does not use intermediate message digests.

2. Digital Envelopes

2.1 Introduction

The signed documents or messages being sent are in unencrypted form which might be unacceptable in situations when extremely sensitive data is being sent. Digital envelopes come to the rescue to ensure the privacy of the data being sent.

2.2 Generation of Digital Envelopes

Digital envelopes use a combination of symmetric key and public key algorithms to ensure the privacy of data being sent. The symmetric key is a one time Data Encryption Key (DEK) generated for encryption. Data is then encrypted using DEK. DEK is then encrypted using the receiver's public key. Thus, the combination of the symmetrically encrypted data and the asymmetrically encrypted DEK is a digital envelope. The process is shown in Fig 3.
2.3 Verification of Digital Envelopes

The receiver, upon receiving the digital envelope, needs to first decrypt the DEK using his/her secret key. This generates the DEK which is then used to decrypt the message. The process is described in Fig 4.
2.4 Digital Envelopes and Digital Signatures

The next step in ensuring the integrity of the data being sent is to digitally sign a message in the digital envelope. This ensures the privacy of the data being sent plus the privacy of data. This is just an extension of the previous protocols with the message now being message plus the signature.

3. Protocols

In this section we describe a cryptographic protocol SET which uses digital signatures for authentication.

3.1 SET

3.1.1 Introduction

SET refers to Secure Electronic Transaction protocol, an open protocol developed by MasterCard, VISA IBM et al. It is the present industry-wide standard for secure online credit card transactions.

SET provides the following services

- Confidentiality of information
- Payment Integrity
- Authentication for merchants and cardholders

SET provides the above services using these features

- Digital certificates
- Trusted third party for authentication
- Hierarchy of trust

3.1.2 Plastic Credentials

To understand the features employed by SET, and how SET uses them to provide the aforementioned services, one needs to understand the concept of "plastic credentials" used in an offline scenario.

One such situation frequently occurs in paper check scenario which a buyer writes to make a purchase with a seller. To verify the authenticity, the buyer presents a driver's license. Although the seller doesn't know it is exactly you, he or she is willing to defer to the Department of Motor Vehicles (DMV) as a trusted third party to vouch for your identity (It is however up to you, to identify yourself to DMV to get the license). The DMV has issued a plastic card identifying you and has "signed" it. (The authenticity of the seller is a non-issue here, but is not in an online scenario). With such a system, the seller relies on a trusted third party, DMV for authentication.

To further carry the analogy, SET uses the following system to provide the services:
However, there is one more complication. Who is the trusted third party that issues the "plastic credentials" here? This is further complicated by the fact that there are many banks or financial institutions that issue credit cards!

### 3.1.3 Hierarchy of Trust

SET uses a Hierarchy of trusted third parties to establish a system of trust. To understand how it works, let us assume a simple scenario of one buyer and one seller who conduct their business through banks Bank1 and Bank 2 as in Fig 5.

The credit card company sits at the top of the hierarchy and vouches for the banks by providing a digital certificate to the banks. The banks, in turn, vouch for the sellers and buyers by issuing digital certificates to them. (As before, it is up to the banks, sellers and buyers to identify themselves to the issuing entity)

### 3.1.4 What is a Digital Certificate?

Up till now, we have been talking about digital certificate without bothering about the details. Now is the time to bare digital certificates.

Digital certificates are documents containing the identity of the seller or buyer of issuing banks (like, card numbers, addresses et al) digitally signed by the issuing entity (like the credit card company or the issuing bank) plus the public key of the seller, buyer or the issuing bank. Since it is a digitally signed document, it ensures the integrity of the data being transmitted and also authentication.
3.1.5 What about Privacy of Transaction?

So far, we have seen how SET provides the services for ensuring integrity of data being transmitted and authentication. But, what about confidentiality of information?

A SET transaction starts by exchange of digital certificates for authentication. Since digital certificates contain public keys, information transmitted will now be encrypted using the public keys. This ensures confidentiality of information.

4. Extension

Digital signatures enable people to affix unique signatures to documents or message that they send. There are however, certain situations where anonymity is required or where a "group" signs a document or message. This introduces extensions of digital signatures: Blind Signatures and Group Signatures.

4.1 Blind Signatures

4.2 Introduction

Blind Signatures were introduced by Chaum as a method of ensuring spender anonymity in digital cash systems. In a blind signature protocol, the signer signs the document or message without knowing the contents of it. Moreover, if the signer ever sees the document/signature pair again in future, he/she can verify that the signature is indeed his/hers but should not be able to identity when or for whom he/she signed it.

4.3 Chaum's Blind Digital Signature Scheme

Chaum's scheme is a variant of RSA algorithm. To see how the scheme works, let us suppose our two friends Alice and Bob and Alice wants a document to be blindly signed by Bob. As outrageous as it might seem now ("Why would anybody sign on something whose contents are unknown?", I hear the scream), we will later see the applicability of the scheme to electronic cash and online voting.

Blind Signature protocol for message M

Alice Round 1

1. Alice informs Bob that she needs a blind signature for message \( m \).
2. Alice picks a random number \( r \in \mathbb{Z}_n^* \) and computes the "blinded" message \( m' \)

\[
m' = m.r^e \pmod{n}
\]

where \( <n,e> \) are Bob's public key.
3. Alice sends $m'$ to Bob.

**Bob Round 2**

1. Bob signs the document using his private key:

   $$s' = m'^d \pmod{n}$$

   Note that:

   $$m'^d \pmod{n} = (m.r^e)^d \pmod{n} = (m)^d r \pmod{n} \quad \text{Since } de = 1$$

2. Bob sends $s'$ to Alice.

**Alice Round 3**

1. Alice takes the signature $s'$ and “extracts” the appropriate signature for $m$ by dividing the signature by $r$.

   $$s = \frac{s'}{r} \pmod{n}$$

2. The pair $m/s$ now represents a valid message/signature pair which Bob can verify if later if presented with the pair.

The most important thing in the protocol is that Bob issued a signature without knowing the contents of the message. This is due to the blinding factor introduced by Alice. But Bob will be able to verify, on seeing the message/signature pair that the signature is indeed his but will not be able to easily identify for whom he signed it.

**4.4 Applications**

We will now describe the applications are applicable to digital cash and online voting scenarios. We only illustrate basic ideas and do not attempt to provide an exhaustive scheme for these scenarios.

**4.4.1 Electronic Cash**

Assume Alice wants to make an online purchase with digital cash.

The basic electronic cash transaction protocol consists of the following steps:

**Withdrawal**

1. Alice creates digital currency coin C. The currency should contain "valid" information to denote a particular value for a coin and other relevant information.

2. Alice takes C, "blinds" it and sends it to the bank to get it signed.
3. The bank signs it and then deducts the amount C from Alice's amount.

Purchase

1. Upon receiving the signed currency, Alice proceeds with “unblinding” it and gives it to an online retailer for purchase.

2. The online retailer, upon receiving the coin C, sends it to the bank to verify the signature. The bank validates the signature, informs the retailer.

3. The bank deposits the coin to the retailer's account.

4. The retailer sends the goods to Alice.

Note that the identity of Alice is unknown to the retailer here unlike in a credit card scenario. Also, the bank won't be able to "trace" the coin C and thus won't know the spending habits of Alice.

4.4.2 Online Voting

Anonymity in online voting systems is of paramount importance for proper functioning of a democracy. We now see how blind signatures ensure anonymity in these kinds of situations.

Lets us assume that there are two entities, our famous friend Alice and the Election Commission (EC) which oversees the election process. Furthermore let us assume that the election happening is a "Recall" one here in Florida which requires the voter to vote "Yes" or "No". The voting protocol would involve the following steps.

Online Voting Protocol

Registration

1. Alice creates ballots B1 and B2 which consist of a random serial number generated, other relevant information and which represent "Yes" and "No".

2. Alice blinds the ballots and sends it to EC.

3. The EC checks to see Alice has not voted before and then signs the ballots and gives it back to Alice.

Voting

1. Alice unblinds the messages. She now has two valid ballots.

2. Alice picks one, encrypts it and sends it to EC.
3. The EC decrypts the ballot, checks the signature, and checks the serial number in its database and then proceeds to record the vote in the database.

4.4.3 Some extensions

The digital cash protocol described above has some obvious drawbacks. For example, what can prevent Alice from generating a document coin C valued $100 billion (!) and get it blindly signed by the bank? What can prevent Alice from double (or more) spending the coin C?

We now present a few modifications to the basic protocol to overcome these drawbacks. Please note that the extensions presented are also "academic" in nature. Exhaustive presentation of extensions is beyond the scope of the presentation.

4.4.3.1 Detecting fraudulent coins/documents

There are two possible ways to prevent fraudulent coins being generated:

1. The bank can have different keys to denote different currency values. This way, Alice will have to have to get the bank signed with a key that corresponds to the currency value.

2. Alice and the bank can engage in a "cut and choose protocol". In this case, Alice prepares n number of coins which are identical except for the serial numbers on them. She then blinds them and sends to the bank for signing. The bank picks all but one coin and asks Alice for blinding values for them. Alice gives the values as requested. If, after unblinding, the bank finds correct values, the bank then proceeds with signing the remaining coin. This way the bank has high assurance that the remaining coin also has the correct value.

4.4.3.2 Detecting double spending

In this case, we can have the bank maintain a universal list of serial numbers already spent. The bank then has to check such a database for each transaction. This scheme however can be expensive as the sizes of databases grow. Further, if Alice responds the coin quickly before the database is updated, detection would not be possible.

5 Group Signatures

5.1 Introduction

Group signatures extend Digital Signatures to multi-party setting. In a group signature scheme, members of a group are allowed to sign on behalf of a group. The signatures are verified using the group public key. Before we present the features of a group signature scheme, we present a few definitions.

**Group member:** A member of the group who can sign on behalf of the group.

**Group public key:** The public key for the group which is used to verify the group signature.
**Group manager:** The member of the group who has certain special privileges which we will be describe in the next section.

We now present some of the features required for a group signature scheme and a simple protocol. An exhaustive mathematical scheme is presented in [4].

### 5.2 Security Features for Group Signatures

1. **Unforgeability:** The group signatures can be issued only by group members and should be impossible to be forged by external members.

2. **Signer Anonymity:** The group members should not be able to determine the signer of a particular document. Only the group manager should be able to ascertain the identity of the signer.

3. **Undeniable Signer Identity:** The group manager should always be able to determine the identity of a signer of any document and should be able to prove it before a judge.

4. **Unlinkability:** Determining the identity of a signer of a document based on two signed documents should be computationally infeasible for any group member except the group manager.

5. **Security against Framing Attacks:** A subset of group members (including perhaps the group manager) should not be able to collude to sign a document on behalf of another group member.

### 5.3 A Simple Protocol

We present a set of steps for generating a group signature.

**SETUP:** A probabilistic algorithm is set up which generates the group public key $K_g$ and a secret key $K_m$ for the group manager.

**JOIN:** An interactive step between the group manager and a new group member Alice which produces Alice's secret key $x$ and membership certificate $A$.

**SIGN:** An interactive step between Alice and an external member Bob to generate signature $s$ on message $m$ using Alice's secret key $s$.

**VERIFY:** A step which on input $(m, s, K_g)$ verifies the signature.

**OPEN:** A step which on input $(m, s, K_m)$ reveals the identity of Alice.
Conclusions

In this presentation, I have attempted to present a broad picture of digital signatures focusing on their applications and restricting the mathematics for lack of time. Interested readers, please consult references and particularly [4] for a more detailed presentation on group signatures and its extension blind group signatures and their applications.
References


