Group Blind Digital Signatures
Cryptography and Computer Security

Lucia Harceková

lh9056@student.uni-lj.si

February 10, 2020

University *of Ljubljana*
Faculty *of Computer and Information Science*

Abstract

The Group Blind Signatures combine the properties of the Group Signatures and Blind Signatures, it is required that a group member signs a document (on the group's behalf) without knowing its content. Our work aims to apply the acquired knowledge from lectures and other literature about Group Blind Digital Signatures into practice to real problems from everyday life. Given are examples for a better understanding of the process that is hidden behind the digital signing.

Keywords: Digital Signature, Blind Digital Signature, Group Digital Signature, Group Blind Digital Signature

Contents

1 Introduction

In modern times, the trend is to solve every single thing electronically to increase speed and simplify work, but there is no need to forget about security. Digital Signatures are used worldwide as cryptographic authentication mechanisms, which additionally provide integrity of signing messages. There exist many kinds of forms and schemes for them, also there are a lot of sectors in which they are used.

Imagine the situation, if we go to the bank to withdraw money and there is only one person (CEO) who can verify our identity to withdraw a certain amount of money, we would spend ages there. We would appreciate if there were more employee who can do that for us instead of CEO with respect for anonymity. In such cases has Group Blind Digital Signatures application.

This paper provides basic information Group Signatures and Blind Signatures and it focuses especially on Group Blind Digital Signatures, which combine the properties of both previous Digital Signatures. Our goal was to make the clearer working of Group Blind Digital Signatures by practical examples from real-life and bring more understanding, wherever they can be used. Our observations are based on the work of Zulkar Amin Ramzan [7].

In the first part, Digital Signatures are described in general, we mentioned their properties and introduce a scheme proposed by D.R. Stinson. In the next two sections, we also pay attention to a brief description of Group Signatures and Blind Signatures, because understanding them helps the top view of the next chapter dedicated to Group Blind Digital Signatures. Then in the final section, we turn our attention mostly on the application of Group Blind Digital Signatures. Their applications include auctions, e-voting, e-banking, electronic cash and much more.

2 The Digital Signatures

The Digital Signatures are considered as the electronic analog of the traditional handwritten signatures, but they are more difficult to forge than the handwritten type. Whitfield Diffie and Martin Hellman were the first who described the concept of a digital signature scheme in 1976, although they only assumed that such scheme existed. It was them who stood behind the solution of how two parties can securely agree on the secret key through the insecure channel without prior communication. They also build up the idea behind Public Key Cryptosystem, but they never gave specific construction of how it can be implemented in practice.

Here come Rivest, Shamir and Adleman, who compiled the first public-key cryptosystem. One of the main applications of public-key cryptography are Digital Signatures, using the private key for signing and public key for verifying the signature.

The main purpose of digital signatures is to enable someone to digitally sign the documents in a secure and efficient manner and enable other authorized people to verify it on an insecure network (e.g. internet). Digital signature practically serves as a confirmation or better-said guarantee of signers, genuine of an electronic document and approval were given by signatory. To secure these things, the digital signature should have the following properties:

- (publicly verifiable) If one of the receivers verifies the given signature to be legitimate, all others also verify the signed message as being true.
- (non-repudiation) When the signer signs certain message (s)he cannot later deny that action.
- (transferable) It means that the signed message (by the signer) can be shown to 'third-party', which can verify, if the signature is legitimate. The third-party can make a copy of it, then show it to another party and convince it that the signer authenticated message.
- (authentication) Signers use their private keys for signing their messages, as a result, the recipients can be sure of the identity of the signer. The huge importance of this property is especially obvious in the finance part. For example, when a bank client decides to cancel the account, he sends his request to the bank. If the bank is not convinced that such a message is sent from an authorized client, it rejects the request.
- (integrity) When verifying the signature, it is checked that the content of the document matches its content when it was signed. If there is a difference, the control of the verification is unsuccessful. In case a message is digitally signed, any change in the message after signature invalidates the legitimate signature. It is a good way how to check if the message has not been altered.

Idea how digital signatures work:
We want to transfer a document from one person (signer) to another (verifier - receiver). The receiver needs to have some evidence that the document is from a trusted source. That is why there are special keys (public and private) thanks to which they can verify data.

Let us make the view about digital signatures more formal. From the previously mentioned properties, the naive process and analogy that digital signatures should in a certain way practically replace handwritten signatures, it is outlined an idea how the procedure should look like. Here is given the definition of the digital signature scheme proposed by D.R. Stinson [8].

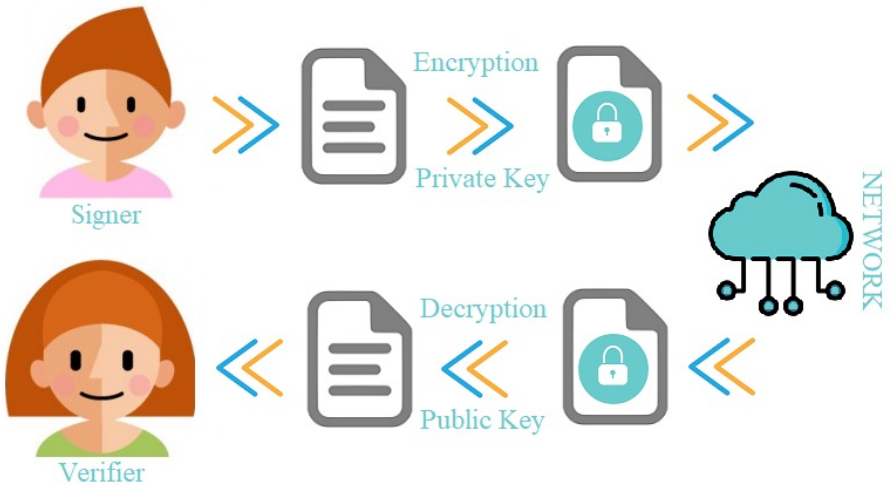

Figure 1: Idea how digital signatures work

Definition 2.1. A signature scheme is a five-tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of possible messages
2. $\mathcal{A}$ is a finite set of possible signatures
3. $\mathcal{K}$, the keyspace, is a finite set of possible keys
4. For each $k \in \mathcal{K}$, there is a signing algorithm $\text{sig}_k \in \mathcal{S}$ and a corresponding verification algorithm $\text{ver}_k \in \mathcal{V}$. Each $\text{sig}_k : \mathcal{P} \to A$ and $\text{ver}_k : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ are functions such that the following equation is satisfied for every message $x \in \mathcal{P}$ and for every signature $y \in \mathcal{A}$:

$$\text{ver}(x, y) = \begin{cases} true & \text{if } y = \text{sig}(x) \\ false & \text{if } y \neq \text{sig}(x). \end{cases}$$

In the next sections are given more information about Blind Digital Signatures, Group Digital Signature, and Group Blind Digital Signatures, but there are many more types such as Multi Signature, Proxy Signature, Ring Signature, ...

3 Blind Digital Signatures

The Blind Digital Signatures are a variant on the traditional digital signatures, however it is used in many applications, which involves anonymity. It allows a participant to sign a message without knowing what the message is. Moreover, if the signer ever sees the document/signature pair, he should not be able to determine when or for whom he signed it, even though he can verify that the signature is valid. It allows the realization of untraceable payments systems that offer improved audibility and control compared to other signatures, while at the same time offering increased personal privacy [3].

In 1982 an American computer scientist and cryptographer David Chaum published the concept of Blind Digital Signatures. The author is known as an inventor of e-Cash, which means an electronic cash application, that aims to preserve a user's anonymity.

Security of these signatures can be proven by the two most efficient ways:

1. complexity-based proofs,
2. random oracle model based proofs.

The blind signatures have to satisfy the following requirements:

- (anonymity) Anonymity prevents the signer from linking later the blinded message to unblinded version, that it may be called upon to verify.
- (verifiability) The receiver (verifier) of the signature is able to verify that signer's private key formed the signature.

3.1 The Blind Digital Signature Scheme

The two well-known blind signature schemes are called the Chaum's, based on RSA public key cryptosystem, and the Blind Schnorr. Schnorr improved the efficiency of Chaum protocol, which enables using it in smart card applications.

Chaum's Blind Digital Signature Scheme - RSA

The implementation of this concept using RSA works as follows. Suppose Anna has a message m that she wants to send to Katka. Katka needs to sign it, but Anna does not want Katka to see information in m. Katka's public key is (n, e) and d is her private key. Anna pick a random variable r such that $\gcd(r, n) = 1$ and sends to Katka.

The value m' is blinded by a random variable $r \in \mathcal{Z}_n$

$$m' = r^e \; m \mod n.$$

So, Katka can not get useful information from it. Then Katka returns the signed value s' to Anna

$$s' = (m')^d = (r^e m)^d \mod n.$$

Anna can get the true signature s of m by computing

$$s = s' r^{-1} \mod n.$$

As you can see, Anna has the signature on the message which is blinded by a random variable.

Blind Schnorr Digital Signature Scheme

Let (g, p, q, y) be the public key, where (g, p, q) describe a group of order q, with a generator g and $y = g^x \mod p$. Let $\mathcal{H}()$ be a hash function, that maps an element in the space $\{1, 2, \ldots, q-1\}$.

Anna wants to send a message m to Katka to sign it. Anna chooses a random number k from 1 to $(q-1)$ and computes the signature r as follows:

$$r = g^k \mod p$$

and sends it to Katka. She picks a random variable a, b in $(1, \ldots, q-1)$ and computes (r', e', e) as:

$$\begin{gathered} r' = r(g^a)(y^b) \mod p, \\ e' = \mathcal{H}(M||r'), \\ e = e' - b \mod q, \end{gathered}$$

and sends e to Alice. Anna computes s

$$s = ex + k \mod q,$$

and sends it to Katka. Katka computes s'

$$s' = s + a \mod q.$$

The resulted pair is a valid signature r', s' on m. As you can see, the Katka can see only e, which is based on a hash of m. So the information Anna sends is blinded by the random factor b [6].

4 Group Digital Signatures

The Group Signatures were introduced by van Heyst and Chaum in 1991 [7]. Comparing to ordinary signatures, signers are in anonymity hidden under certain group. By verification, it can be only said that the message or document was signed by a member of certain group. However, someone with the role of "group manager" can in special cases, if it is needed ,"open" signature and reveal the true identity of the signer.

The Group Signatures are often used in companies where the employees represent the entire corporation. For example, if we go to the bank to withdraw money and there is only one person (CEO) who can verify our identity, we would spend ages there. In this case, the CEO can set up a group signature scheme, and act as the group manager. Specific employees can validate and sign documents or messages on behalf of the entire company. The clients would only have to use a single company public key to verify the signatures, they aren't able to identify which employee has signed the document. Only the group manager can do that.

Some of the characteristics of the Group Digital Signatures may already be apparent from the above example and the previous definition, let us summarize them:

- (unforgeability) The signature that is verifiable by the group public key can be produced only by group members.
- (coalition resistance) No subset of group members should be able to generate valid group signatures that are untraceable. In this way, we can prevent attacks in which a coalition of group members get together, combine their information and generate signatures that are evaluated to be legitimate, but for which the Open procedure fails to reveal any group member.
- (unlinkability) It is computationally impossible for everyone but the Group Manager, to determine if two different signatures were computed by the same group member.
- (security against framing attacks) No subset of group members can sign a message on behalf of another group member.
- (undeniable signer identity) The identity of the group member who issued a valid signature can be always determined by the group manager. He can also prove to some other entity, which member signed a given document without compromising that particular group member's anonymity in previous or future messages he may sign.
- (conditional signer anonymity) Anyone can check if a message signature pair was signed by some group member, but only the group manager can determine, which specific member issued the signature.

When we talk about the group, in general, we do not mention its parameter which has an impact on efficiency. Particular parameters [7]:

- The size (number of bits) of the group public key γ.
- The size (number of bits) of an actual group signature on a message.
- The efficiency of the Signing, Verification, Initial setup, Opening, and Joining.

Let us now notice one more definition about the description of procedures used to realize the Group Signature designed by Ateniese, Camenisch, Joye, and Tsudik [2].

Definition 4.1. The group signature scheme is a digital signature scheme comprised of the following five procedures:

`SETUP:` On input a security parameter k, this probabilistic algorithm outputs the initial group public key γ (including all system parameters) and the secret key S for the group manager.

`JOIN:` A protocol between the group manager and a user that results in the user becoming a new group member. The user's output is a membership certificate and a membership secret.

`SIGN:` A probabilistic algorithm that on input a group public key, a membership certficate, a membership secret, and a message m outputs group signature of m.

`VERIFY:` An algorithm for establishing the validity of an alleged group signature of a message with respect to a group public key.

`OPEN:` An algorithm that, given a message, a valid group signature on it, a group public key and a group manager's secret key, determines the identity of the signer.

4.1 The Group Digital Signature Scheme

Jan Camenisch and Markus Stadler are swiss research scientists in cryptography, who invented new group signatures: the basic group signature scheme and the efficient scheme. This paper provides only details about the basic scheme based on the work of Camenisch and Stadler.

For these definitions, the reader should be already familiar with terms such as discrete logarithm, double discrete logarithm, e-th root of the discrete logarithm and signature of knowledge, if not just quickly check the appendix with a list of definitions (Appendix 1). In next definitions a signature of knowledge of a double discrete logarithm is marked as SKLOGLOG and signature of knowledge of a n e-th root of discrete logarithm is marked as SKROOT.

`SETUP:`
The group manager computes the following values:

- an RSA public key (n, e) and private key d, where $n = pq$, for $p = 2P + 1$ and $q = 2Q + 1$ and $p,\ q,\ P,\ Q$ are all primes
- a cyclic group $G = \langle g \rangle$ of order n in which computing discrete logarithms is hard
- an element $a \in \mathbb{Z}_n^*$, for which has large multiplicative order modulo for both prime factors of n
- an upper bound λ on the length of the secret keys and a constant $e > 1$ (these parameters are required for the SKLOGLOG signatures)

The group's public key is $\gamma = (n,\ r,\ G,\ a,\ \lambda,\ e)$.

`JOIN:`

When Alice wanna join the group, she chooses her secret key $x \in_R \{0, \ldots, 2^\lambda - 1\}$ and calculates the value $y = a^x \ (mod\ n)$ and her membership key $z = g^y$. She commits herself to y, for instance by signing it. She then sends $(y,\ z)$ to the group manager and proves to him that she knows the discrete logarithm of y, denote it as x, to the base a. When the group manager is convinced that Alice knows x, he gives her the membership certificate $v \equiv (y+1)^d \ (mod\ n)$. It should be not possible to construct (x, y, v) without the help of the group manager: if y is correctly formed then it is infeasible to compute the e-th root of $y + 1$ because the factorization of n is unknown. On the other hand, if $y + 1$ is computed as w^e for some vale w it is infeasible to compute the discrete logarithm of $w^e - 1$ to the base a. Furthermore, even if several group members pool their values, they still seem unable to construct a new such triple.

`SIGN:`
For signing the message $m \in \{0,\ 1\}^*$, Alice computes the following values:

- $\widetilde{g} = g^r$ for $r \in_R \mathbb{Z}_n^*$
- $\widetilde{z} = \widetilde{g}^y$
- $V_1 = \text{SKLOGLOG } [\alpha : \widetilde{z} = \widetilde{g}^{a^\alpha}](m)$
- $V_2 = \text{SKROOTLOG } [\beta : \widetilde{z}\widetilde{g} = \widetilde{g}^{\beta^e}](m)$

The signature on the message m is $s = (\widetilde{g},\ \widetilde{z},\ V_1,\ V_2)$.

`VERIFY:`
To verify the given signature s on the message m. The verifier needs to do:

- check if V_1 is the valid signature of knowledge of the double discrete logarithm
- verify that V_2 is valid signature of knowledge of the root of the discrete logarithm

V_1 shows that $\widetilde{z}$ is equal $(\widetilde{g})^{a^\alpha}$ and thats why we know that $\widetilde{z}\widetilde{g}$ has form $\widetilde{z}\widetilde{g} = (\widetilde{g})^{a^\alpha+1}$, where α is some integer which is known to the signer. And V_2 proves that the signer knows e-th root of the logarithm of $\widetilde{z}\widetilde{g}$. Both conditions together show that the signer knows α and $(a^\alpha + 1)^d$, this corresponds to the membership certificate and the secret key.

`OPEN:`
We have two signatures $(\widetilde{g},\ \widetilde{z},\ V_1,\ V_2)$ and $(\widetilde{g}',\ \widetilde{z}',\ V_1',\ V_2')$, deciding if these two signatures were issued by the same group member, means computing: $\log_{\widetilde{g}} \widetilde{z} = \log_{\widetilde{g}'} \widetilde{z}'$. Solving this problem is infeasible and therefore are the signatures of the group members anonymous. However, the group manager knows the relatively few possible values of $\log_{\widetilde{g}'} \widetilde{z}'$, namely the discrete logarithms (to the base g) if the membership keys of the group members, and therefore perform this test. Given only a signature $(\widetilde{g},\ \widetilde{z},\ V_1,\ V_2)$ for the message m. he can find the group member who issued this signature by testing: $\widetilde{g}^{yP} = \widetilde{z}$, for all group members P.

There is a lot of different schemes, the difference can be in the way of providing anonymity, adding members, way of opening signature ... However, this issue is not the main theme of this work and the previous information about group signatures covers enough details to understand the next chapters. However, if you are more interested in this subject, it is recommend the book *Group Signatures: Authentication with Privacy* [11].

5 Group Blind Digital Signatures

The Group Blind Digital Signature, even according to its name, is a combination of Blind Digital Signatures and Group Digital Signatures. Properties of the Group Blind Digital Signature have related to these of the Group Digital Signature mention in the previous section 4. Moreover, from the Blind Digital Signature, it brings anonymity to the context of signing a message. This property is in lots of texts known as the Blindness of Signatures.

- (blindness of signatures) The signer is not familiar with the content (specific information in) of the message he signs. And the signer is unable to memorize signing of a certain message, but he can verify if the signature is valid.

Also, the procedures given in Blind Group Digital Signatures (joint, setup, verify, sign and open) are identical to those given in the Group Digital Signature model.

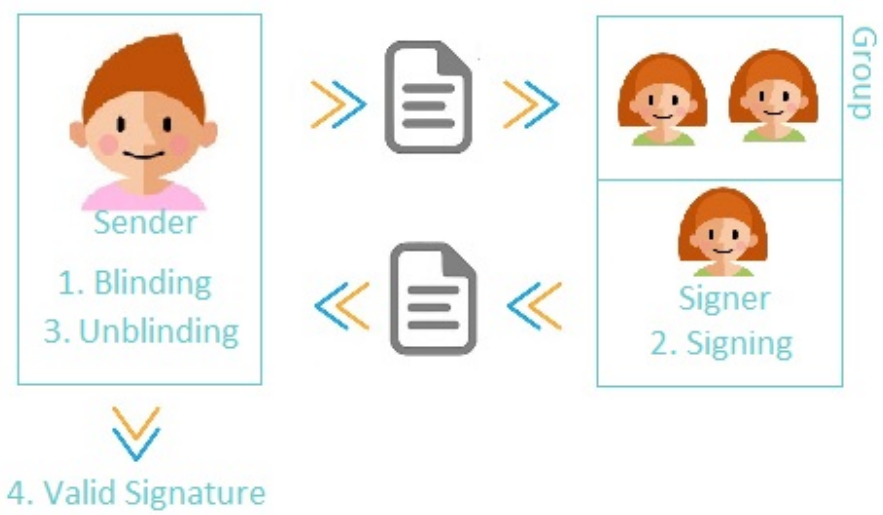

Figure 2: Idea how Group Blind Digital Signatures work

5.1 The Group Blind Digital Signature Scheme

Let us turn our attention to what the Group Blind Digital Signature scheme looks like. As it was said before, Group Blind Digital Signatures have a lot in common with Group Digital Signatures. Therefore we will improve scheme based on the work of Camenisch and Stadle (Section 4.1) and use it for Group Blind Digital Signature Scheme.

The parts: Join, Open, Verify, Setup are the same. The only difference is in the way of signing the message (protocol Sign). This part needs to be changed because we want to hold the property taken from Blind Digital Signature and let signing message m to be blind.

`SIGN:`
The user wants signer to sign a message m, Alice does the following steps:

1. Receive $q \in_R \mathbb{Z}_n^*$ and let $\tilde{g} = g^q$, $\tilde{z} = \tilde{g}^y$.
2. Obtain random u_i, such that
   $2^\lambda \leq u_i \leq 2^{\lambda+\mu} - 1$ $(1 \leq i \leq \ell)$
   and set
   $P_i^{SKLOGLOG} = \tilde{g}^(a^{u_i})$ $(1 \leq i \leq \ell)$.

3. Then obtain random $v_i \in \mathbb{Z}_n^*$ $(1 \leq i \leq \ell)$ and set
$P_i^{SKROOTLOG} = \widetilde{g}^(v_i^e)$ $(1 \leq i \leq \ell)$.

4. Finally, signer sends $(\widetilde{g},\ \widetilde{z},\ \{P_i^{SKLOGLOG}\},\ \{P_i^{SKROOTLOG}\})$ to the user.

Now, the user performs the following process:

1. Obtains $b \in_R \{1, \ldots, 2^{\lambda-1}\}$ and $f \in_R \mathbb{Z}_n^*$ and sets $w = (af)^{eb}\ (mod\ n)$.

2. Further sets:
$\widetilde{g} = \widetilde{g}^w$,
$\widetilde{z} = \widetilde{z}^w$,
$\widetilde{P}_i^{SKLOGLOG} = (P_i^{SKLOGLOG})^w$,
$\widetilde{P}_i^{SKROOTLOG} = (P_i^{SKROOTLOG})^w$.

3. Take $\{\widetilde{P}_i^{SKLOGLOG}\}$ as the commitment values. Adjust the responses $\{t_i^{SKLOGLOG}\}$ by adding eb.

4. Take $\{\widetilde{P}_i^{SKROOTLOG}\}$ as the commitment values. Adjust the responses $\{t_i^{SKROOTLOG}\}$ by multiplying by $(af)^b$.

In the end user has:

- $V_1 = SKLOGLOG[\alpha : \widetilde{z} = \widetilde{g}^{a^\alpha}](m)$
- $V_2 = SKROOTLOG[\beta : \widetilde{z}\widetilde{g} = \widetilde{g}^{\beta^e}](m)$

The resulting signature on the message m is $s = (\widetilde{g},\ \widetilde{z},\ V_1,\ V_2)$.

And how we know that the signature s on message m is really blind? The signer's input was blinded by changing $\widetilde{g}$ and $\widetilde{z}$ into random $\widetilde{g}$ and $\widetilde{z}$ by random blinding factor w. And two constructed signatures of knowledge are blind. And that is the reason why the signature cannot be linked to the signer's view of the protocol.

Maybe it looks like we did not put enough attention to Group Blind Digital Signatures. But the thing is, that it contains a lot of stuff from Group Digital Signature and Blind Digital signature, and there is no reason for repeating what was already mentioned. Now we can fluently continue to examples, which should help us to see the connection between the previous sections.

6 Examples

There are various applications of Group Blind Digital Signature. In this chapter, it is shown the basics process hidden behind real-life situations, where the Group Blind Digital Signatures are used. The aim was to avoid the enormous number of formulas and fully concentrate on situations. Because many articles contain especially only naked formulas, which can be for new readers a little bit chaotic and don't give them a really good reason for why they should continue to read and learn more about this topic. My motivation was to bring the abstraction layer from the given pattern to help newbie not to get lost and see why Blind Group Digital Signatures are really useful.

Note: For making the life of the user easier, there are many applications that practically replate the role of the user mentioned in some of the following examples. The user lets this application perform operations instead of him and allows to it use his information (ex. through user interface user clicks on button submit, write password or ID, ...).

6.1 Referendum

Assume that we have a referendum in the city. We do not want to have 666666 positive answers in the city where only 1666 people are living and definitely do not want to let somebody vote repetitive. That is why the Group Blind Digital Signature takes part in the modern way of voting. It is really important to verify the identity of the user and oversee how many times he has already voted. Usually, these things are to deal directly with the selection office[1]. People who come there must at first identify themself with their IDs, then they come behind the prepared temporary wall and select from option, for our case, referendum got options "yes" or "no" (each of the residents has only one vote and all votes are equal). And our final question, how can we deal with this digitally?

At first, it is important to realize who are participants in this case. We have voter Anna, some kind of the registered office and the State Authorized Office to process votes (STO). Next, we can simulate the real process of registration and voting digitally thanks to the Group Blind Digital Signature as follows:

Registration:

1. Anna creates two ballots[2] A_1 (vote "Yes") and A_2 (vote "No"). These ballots include information such as a serial number, other information connected with voting.

2. For making these ballots acceptable as valid, firstly Anna blinds them and sends them to the registered office.

3. The registered office confirms if Anna hasn't voted before (database). If not, then it signs the blinded ballots and returns them to Anna.

4. Anna unblinds returned blinded ballots. And receive two valid votes signed by the registered office.

[1] The place where residents come to vote. At first, they identify themself with valid IDs and then put their vote into the box.

[2] A ballot is a device used to cast votes in an election, and may be a piece of paper or a small ball used in secret voting.

Voting:

1. Now, she chooses her vote ("yes" or "not") and encrypts it with the registered office's public key.
2. Then she sends it to the State Authorized Office to process vote.
3. The State Authorized Office to process vote (STO) decrypts the vote checks if the signature of the vote is valid by using the registered office's public key, also it checks if identification number on the vote has not been used before (database). If everything is alright, STO adds the serial number of the vote to the database and tabulates the vote.

6.2 Shopping online

We now outline how to apply the Group Blind Signature idea to digital cash. It is pretty common today to buying things online. Take this case, young lady want to buy a dress in an online shop. Make our example easier let's assume that young lady and online shop use the same bank. The entire process can be divided into three parts: withdrawal, spending, and deposit.

Withdrawal:

1. The young lady combines the amount that she is going to pay with bits, which specify information such as a large serial number to digital currency into the electronic coin C.
2. The lady asks bank to sign the coin C.
3. The young lady and the bank perform the signature protocol (blinded). Bank applies the signature to C.
4. After successful completion of the previous step, the bank brings down the money from the young lady bank account.
5. Now, the young lady has a valid signature for currency C.

Spending:

1. The young lady takes valid currency C and the bank's signature on currency and gives it to the Online shop.
2. The online store checks if the currency C is valid by the Bank's public key. If the signature is invalid, the protocol is immediately ended. Otherwise, next step.

Deposit:

1. The online store has now digital currency and Banks's signature on currency, both gives to its Bank (BankOS).
2. The BankOS verifies if the signature on the currency is valid. If it is valid the BankOS needs to check if the currency has been already used.
3. If all checks out then Online Shop gives the young lady the requested dress.

The interesting part is that only a young lady's identity stays in anonymity for the bank but also for the online shop. Notice, when the transaction is complete, it is impossible to track any customer since identity is anonymous.

6.3 Electronic cash

The banks are covered by the central bank. Banks together create the group and the group manager is the central bank. Also, the customers of the bank create the group. Because if we want to open an account they need some special identifier (the private key associated with the bank's public key) and later when they gonna pay something they want to stay in anonymity under the group of all customers.

Withdrawal:

1. Anna wants to withdraw money, she needs at first create electronic coin denoted as C. As in the example before, C consists of the serial number and other information.
2. Anna requests her bank for signing the C.
3. Anna's bank applies Group Blind Digital Signature to C and withdraws money from Anna's account. Anna now holds C as well as a valid signature from the bank on C.

Spending:
Anna wants to spend the money on coffee in Teddy's shop. The process:

1. Anna gives the bank's signature and C to Teddy.
2. Teddy verifies if the bank's signature on C is valid (by the bank's group public key). If the signature is not valid the operation is ended. Otherwise, Teddy generates a random sequence of bits c. Then he creates the message containing c, current time, his identity and sings it with his private key (associated with the bank). After that, he sends the message and signature to Anna.
3. After receiving the message from Teddy. Anna checks if the Teddy's signature is valid and if the time in the message is recent. Then, she constructs the message with C, the bank's signature on C and Teddy's message, she signs this message with her group signature and sends it back to Teddy.
4. Now, Teddy received the message from Anna composed from C, signature on C and his message. If the message has a bad form, the process ends. Else he checks Anna's group signature on the message. If Everything is OK, Teddy gives Anna coffee <3.

Deposit:

1. Teddy takes the C, Anna's signature and bank's signature on Cm, and gives it to his bank (Tbank).
2. BankT needs to verify C by checking the signature on C and also checks if the C has been already spent somewhere. If the coin was already spent, the process is ended (the bank owns procedures for dealing with fraud). But if everything is all right bankT credits money on Teddy's account.

6.4 Auction

The principle of the auction is that someone offers an item for sale and others submit bids, they always try to outbid each other. The winner is the one who offered the highest amount of money after a given time period, where no higher bid than the current one has not been made. The winner pays the seller an amount of the money equal to his bid and the seller gives the winner the item. E-auction is an electronic method of a traditional auction.

To make e-auction work (naive version) we assume that we have a group of people who can submit bids called buyers, the group manager is system authority denoted SA. There is also a list of items I and a table denoted T, which is a public broadcast channel with memory. Only SA can write to T and no one else can erase any information from it.

Submit a bid:

1. A buyer selects an item i out of possible items I, for which he wants to submit a bid. And also choose the bid, it must be higher than the current value.
2. The buyer signs i with Blind Signature with his private key (associated with group public key) and sends i to SA.
3. SA verify that it came from an authorized member (group public key). If everything is alright, then SA actualizes value in table T. SA also returns proof of receipt for the submitted item i. This proof is used in the Claiming phase.

This process is repeated until the end of the given time period for auction (no higher bid than the current one has not been made).

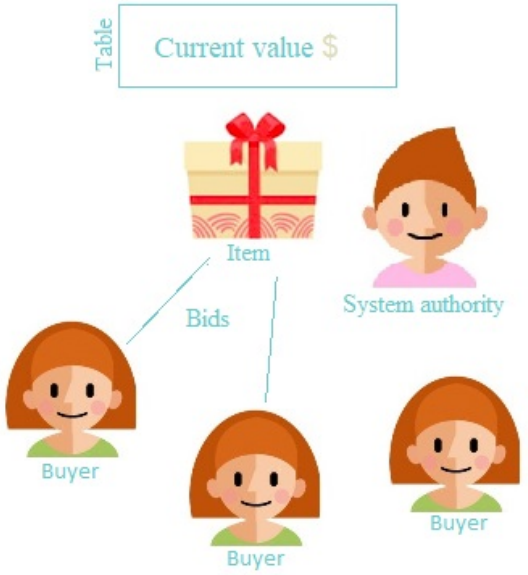

Figure 3: The idea of the e-auction process

Result:
After the end of the phase Selecting item. SA declares the winner for item i by opening the signature. Thanks to the properties of the signature buyer cannot just change his mind and hide, but it can be verified that he is the one who submitted the winner bit (SA record).

6.5 Other

- **Printers:**
  The firm has a system of printers for different departments. Only a user from a certain department can use the printers in that department. The user must identify himself before using the printer. The company respects privacy protection so it does not reveal the identity of the user who is printing. But someone spent a lot of toner, the group manager can find out the name of the villain.

- **Keycard access:**
  Some parts of the building are accessible only to authorized people (members of the group). However, it is not desired to track people's movements.

- **Submit tenders:**
  All companies which submitting the tender form a group. When signing the tender anonymously (group signature). Only after the result, the winner can be traced, but others stay still in anonymity.

- **Electronic test:**
  The teacher wants to give students an online test from mathematic, only students who visited his classes (the group) should be able to sign and fulfill the test. In order for the teacher to be objective in correcting the tests, students are in anonymity until the all test are corrected.

7 Conclusion

This paper-work brings a brief study of digital signatures, their properties, and their purpose. The Group Digital Signatures and Blind Digital Signatures are also shortly summarized to bring a better overview. The main attention was given to Group Blind Digital Signatures and their applications. The intention of writing was to provide a better understanding of how can the digital signatures specifically the Group Blind Digital Signatures be used in real-life situations, for example, we succinctly brushed up on the topic of online shopping, voting or it was also mentioned e-banking and others.

1 Supplementary definitions

Let G be a cyclic group of order n generated by some $g \in G$ (hence $G = \langle g \rangle$) a note $a \in \mathcal{Z}_n^*$.

Definition 1.1. The discrete logarithm of $y \in G$ to the base g is such integer x satisfying

$$g^x = y.$$

Definition 1.2. The double discrete logarithm of $y \in G$ to the bases g and $a \in \mathcal{Z}_n^*$ is an integer x satisfying:

$$g^{(a^x)} = y$$

if such an x exists [9].

Definition 1.3. An e-th root of the discrete logarithm of $y \in G$ to the base g is an integer x satisfying

$$g^{(x^e)} = y$$

if such an x exists [9].

Definition 1.4. An $(l+1)-$ tuple $(c, s_1, \ldots, s_l) \in \{0,1\}^l \times Z_n^l$ satisfying

$$c = \mathcal{H}_l\left(m||y||g||g^{s_1}y^{c[1]}||g^{s_2}y^{c[2]}|| \ldots ||g^{s_l}y^{c[l]}\right)$$

where $c[i]$ is the i -th leftmost bit of c, is a **signature of knowledge of the discrete logarithm** of $y \in G$ to the base g on a message m, with respect to security parameter l, denoted

$$SKLOG_l\left[\alpha | y = g^\alpha\right](m) \text{ [7]}.$$

Definition 1.5. An $(l+1)$-tuple $(c, s_1, \ldots, s_l) \in \{0,1\}^l \times \mathcal{Z}^l$ satisfying the equation

$$c = \mathcal{H}_l\left(m||y||g||a||t_1|| \ldots ||t_l\right), \text{ where } t_i = \begin{cases} g^{(a^n)} & \text{if } c[i] = 0 \\ y^{(a^n)} & \text{otherwise} \end{cases}$$

is a **signature of knowledge of a double discrete logarithm** of y to the bases g and a, and is denoted $SKLOGLOG_l[\alpha : y = g^{(a^\alpha)}](m)$ [10].

Definition 1.6. An $(l+1)$-tuple $(c, s_1, \ldots, s_l) \in \{0,1\}^l \times \mathcal{Z}^l$ satisfying the equation

$$c = \mathcal{H}_l\left(m||y||g||a||t_1|| \ldots ||t_l\right), \text{ where } t_i = \begin{cases} g^{\left(s_i^e\right)} & \text{if } c[i] = 0 \\ y^{(s_i{}^e)} & \text{otherwise} \end{cases}$$

is a **signature of knowledge of a n e-th root of discrete logarithm** of y to the base g, and is denoted $SKROOTLOG[\alpha : y = g^{(a^{\alpha^e})}](m)$ [10].

Bibliography

[1] Z.A. Ramzan, Group Blind Digital Signatures: Theory and Applications, Master of Science, MIT, 1999. http://groups.csail.mit.edu/cis/pubs/ramzan/ramzanms.ps

[2] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. Department of Information and Computer Science, University of California, Irvine, CA 92697-3425, USA.

[3] David Chaum, Blind signatures for untraceable payments. In Proc. CRYPTO 82, New York, 1983. Plenum Press.

[4] D. Pointcheval and J. Stern, New Blind Signature Equivalent to Factorization. Proc. of the 4th CCCS, ACM press. 1997

[5] A. Juels, M. Luby and R. Ostrovsky, Security of Blind Signatures. Proc. of Crypto 97, LNCS 1294, Springer Verlag 1997.

[6] *A note on blind signature schemes*, https://blog.cryptographyengineering.com/a-note-on-blind-signature-schemes/

[7] J. Camenisch and M. Stadler, Efficient group signatures for large groups. In Proc. CRYPTO 97, Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1294.

[8] D. R. Stinson. *CRYPTOGRAPHY: Theory and Practice*. CRC Press. 1995.

[9] S.Burton and J. Kaliski, Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. 1997.

[10] J. L. Camenisch, Advances in Cryptology - CRYPTO '97. RSA Laboratories, 20 Crosby Drive, Bedford, MA 01730-1402 USA, 1998.

[11] M. Manulis, N. Fleischhacker, F. Gunther, F. Kiefer and B. Poettering, Group Signatures: Authentication with Privacy. Cryptographic Protocols Group, Department of Computer Science, Technische Universitat Darmstadt, GERMANY, 2012.