Identity Based Blind Signature

Elham Soleimani, Yasamin Vaziri

September 29, 2024

Contents

1 Introduction **2**

2 Identity Based Encryption **2**
2.1 Definiton . . . 2
2.2 History . . . 2
2.3 Overview of cryptographic operations . . . 3
2.4 Pros and Cons . . . 3

3 Blind Signature **3**
3.1 Definition . . . 3
3.2 Security . . . 4
3.3 Usages . . . 4

4 Identity Based Blind Signature **4**
4.1 Scheme 1[6] . . . 5
4.1.1 Correctness of the Protocol . . . 6
4.2 scheme 2 [2] . . . 7
4.3 Comparison . . . 7
4.4 Test Result . . . 9

5 Electronic voting with IBE [2] **11**
5.1 Overview . . . 11
5.2 Mahender Kumar E-voting System Design [2] . . . 11
5.2.1 Registration . . . 11
5.2.2 Authentication . . . 12
5.2.3 Vote casting . . . 12
5.2.4 Vote counting . . . 12
5.2.5 Security Analyse . . . 13
5.2.6 Security Challenges . . . 13
5.3 Our Contribution . . . 13
5.3.1 new Vote casting . . . 13
5.3.2 new Vote counting . . . 14
5.3.3 Conclusion . . . 14
5.4 Comparison . . . 14

6 Conclusion **15**

Abstract

This report explores the implementation and security implications of Identity-Based Encryption (IBE) and Blind IBE, focusing on their applications in secure e-voting systems. By integrating the framework of IBE with the anonymity properties of blind signatures, a novel cryptographic solution that enhances voter privacy and ballot security is proposed. The practical and theoretical aspects will be examined, demonstrating the scheme's robustness against common cryptographic attacks and its feasibility for real-world application.

Keywords

Identity-Based Encryption, Blind IBE, E-Voting, Cryptography, Blind Signature, Security, Privacy

1 Introduction

Firstly we introduce Identity-Based Encryption and then we explain Blind signature and its practical and cryptographic usage. We describe two Blind signatures based on IBE. Finally, we explain an e-voting system based on the blind signature that protects voter privacy and works even against powerful computers (quantum computers) using Identity-Based Blind Signatures. A trusted body issues digital identities. Voters can delegate their vote to someone else (proxy) while keeping their vote secret (blind signature). The system is secure because it relies on a difficult math problem. This allows for secure multi-region elections where people can vote remotely without revealing who they voted for.

2 Identity Based Encryption

2.1 Definiton

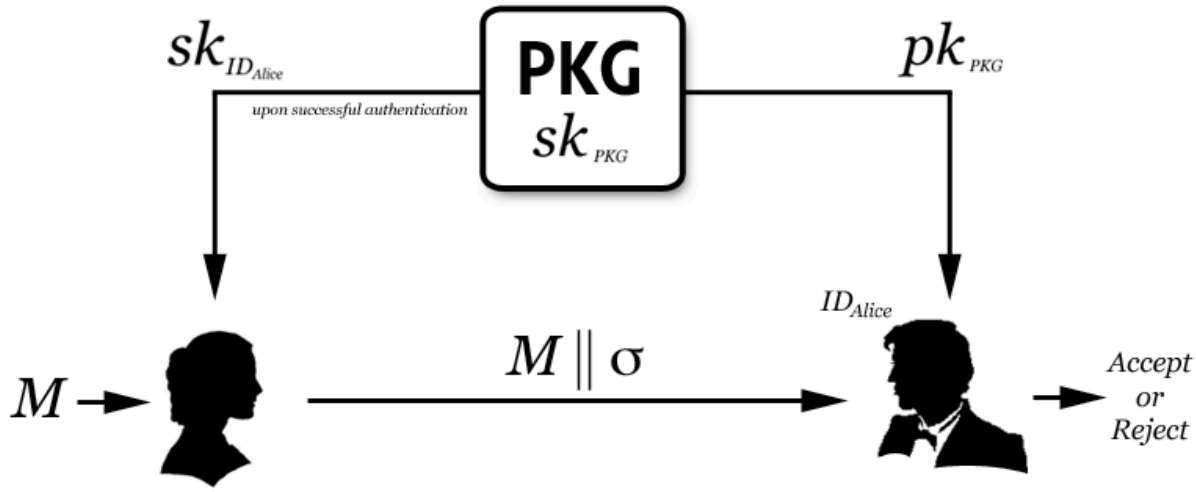
Its reliance on a shared public-key infrastructure (PKI) hinders public-key cryptography's adoption. Secure communications require both parties to generate keypairs, submit certificate requests with identity proof to a Certificate Authority (CA), and obtain CA-signed certificates for mutual authentication and encrypted message exchange. This process is time-consuming, error-prone, and particularly challenging for novice users. Many users capable of receiving encrypted emails struggle to send secure messages due to inadequate preparation, limited interoperability, device constraints, or technical incompetence. Consequently, sensitive communications often occur without encryption. Identity-based cryptography (IBC) offers a solution by eliminating the recipient's preparatory requirements, thereby simplifying secure communication. However, IBC also presents its own set of challenges.

2.2 History

In 1984, Adi Shamir introduced identity-based cryptography (IBC), which uses user identity attributes instead of digital certificates for encryption and signature verification, simplifying cryptographic systems. The challenge of identity-based encryption (IBE) was solved in 2001 by Boneh, Franklin, and Cocks, leading to extensive research and commercial products like those from Voltage Security, Inc. IBC reduces complexity and facilitates secure communication for unprepared users.

2.3 Overview of cryptographic operations

- Alice prepares a plaintext message M for Bob. She uses Bob's identity ID_{Bob} and the PKG's public key pk_{PKG} to encrypt M, obtaining ciphertext message C. Alice then sends C to Bob.
- Bob authenticates with the PKG, essentially sending it sufficient proof that ID_{Bob} belongs to him, upon which the PKG transmits Bob's private key $skID_{Bob}$ to him over a secure channel.
- Bob decrypts C using his private key $skID_{Bob}$ to recover plaintext message M.



2.4 Pros and Cons

- The recipient requires no preparation to receive an encrypted message. This is arguably the most compelling feature of IDC.
- No need to manage a public key infrastructure, including CRL management.
- No PKI means less public information about your enterprise needs to be revealed to those who do not need to know.
- The primary disadvantage of identity-based cryptography (IBC) is its inherent key escrow property, which allows the system to recover a user's encrypted data if their private key is lost but limits user choice and non-repudiation. Unlike PKI systems, which do not escrow signature keys, IBC's key escrow can be a security concern. Variants like certificate-based encryption, secure key issuing, and certificateless cryptography are being developed to address these issues. For instance, secure key issuing distributes master keys across multiple PKGs to enhance security, though at the cost of system performance.

3 Blind Signature

3.1 Definition

The Blind signature was first proposed by Chaum [1] in 1983. Since then several blind signatures have been proposed based on several number-theoretic hard problems like integer factorization, discrete logarithmic problems, etc. Blind signatures are an extension of digital signatures which provide privacy by allowing a user to obtain a signature from a signer on a message without the signer being

able to see the contents of the blinded message. If the signer is later presented with the signed document he cannot relate it either to the signing session or to the user on behalf of whom he has signed the message.
A blind signature scheme is a tuple of polynomial-time algorithms KeyGen, Sign, Verify such that:

- **KeyGen** is an algorithm that on the input of a security parameter, outputs a pair of keys p_k and s_k.
- **Sign** is an interactive protocol between a signer S and a user U. The input of S is a secret key s_k, whereas the input of U is a public key pk and a message $m \in M$ from a message space M. The output of S is a view v (seen as a random variable) and the output of U is a signature γ.
- **Verify** is a verification algorithm that outputs 1 if γ is a valid signature and 0 otherwise.

3.2 Security

Security in blind signatures is captured by two concepts: blindness and unforgeability. Blindness prevents a malicious signer from learning information about a user's message. On the other hand, unforgeability ensures that each completed Sign execution yields at most k signatures after k interactions between S and U.

3.3 Usages

- **Anonymous transactions**: There are several proposals, for instance, Yi—Lam [5](a blind ECDSA scheme for bitcoin transaction anonymity), where blind signatures are used to allow sender anonymity. The use of blind signatures in this proposal indicates that using blind signatures would not require particular modifications besides the obvious substitution of the digital signature scheme: Yi—Lam simply substitutes ECDSA with the blind analog scheme.
- **Anonymous mixing services**: Similarly to anonymous transactions, blind signatures, in combination with mixers, led Valenta and Rowan to introduce Blindcoin [4] built upon Mixcoin with a modification of the protocol to provide anonymity by using blind signatures. Blindcoin guarantees that the input-output connection is kept secret from the mixing service.
- **E-Voting**: Blind signatures are utilized in electronic voting systems to allow voters to submit their votes anonymously. The election authority can verify that the vote comes from a registered voter without knowing which candidate the voter selected, ensuring voter privacy and preventing voter coercion or vote selling.
- **Digital Cash**: Systems like David Chaum's DigiCash used blind signatures to create a digital currency that users can spend anonymously. When a user withdraws digital cash from a bank, the bank signs a blinded version of the currency. The user can then spend this currency without the bank being able to trace it back to the user.

4 Identity Based Blind Signature

Identity-based signatures were introduced by Shamir in 1984 and gave an alternative to prominent Public Key Infrastructure. An identity-based blind signature allows users to interact directly with the signer without any prior interaction with a trusted authority. The first IDBS was proposed in

2002 and several schemes have been proposed since then. An identity-based blind signature scheme is a digital signature scheme that allows a user to obtain a signature on a message without revealing the message to the signer, while the signer's public key can be derived from an arbitrary string (the user's identity) rather than from a certificate. The scheme consists of four main algorithms.

4.1 Scheme 1[6]

Initialization:

1. Let G be a GDH (gap Diffie-Hellman) group of prime order q
2. Let P be a generator of G.
3. Choose a random number $s \in \mathbb{Z}_q^*$ as the master key
4. Compute $P_{pub} = sP$

Setup

1. Hash Functions:
   (a) Define hash functions $H : \{0,1\}^* \to \mathbb{Z}_q$ and $H1 : \{0,1\}^* \to G$
2. System Parameters:
   (a) The system parameters are params $= \{G, q, P, P_{pub}, H, H1\}$
3. Trusted Authority (TA):
   (a) The TA holds the master key s
   (b) For an identity ID, the TA computes the public key $Q_{ID} = H1(ID)$ and the private key $S_{ID} = sQ_{ID}$

Now let see how the Blind Signature Issuing Protocol works:

1. User Side:
   (a) Select a random $r \in \mathbb{Z}_q^*$ and compute $R = rP$
   (b) Send R to the signer.
2. Signer Side:
   (a) Receive R from the user
   (b) Choose random $a, b \in \mathbb{Z}_q^*$
   (c) Compute $t = e(bQ_{ID} + R + aP, P_{pub})$
   (d) Compute $c = H(m,t) + b \mod q$
   (e) Send c back to the user
3. User Side:
   (a) Receive c from the signer
   (b) Compute $S = cS_{ID} + rP_{pub}$
   (c) Send S back to the user

4. User Side (Unblinding):
   (a) Compute $S' = S + aP_{pub}$
   (b) Compute $c' = c - b$
   (c) The user outputs the blind signature (m, S', c')

Signature Verification
To verify the signature (m, S', c'):
Check if $c' = H(m, e(S', P)e(Q_{ID}, P_{pub})^{-c'})$.

4.1.1 Correctness of the Protocol

We derive the verification process as follows:
1. $S' = S + aP_{pub}$
2. $S = cS_{ID} + rP_{pub}$
3. $c' = c - b$

Starting from the verification equation:

$$c' = H(m, e(S', P)e(Q_{ID}, P_{pub})^{-c'})$$

Substitute S':

$$c' = H(m, e(S + aP_{pub}, P)e(Q_{ID}, P_{pub})^{-c'})$$

Substitute S:

$$c' = H(m, e(cS_{ID} + rP_{pub} + aP_{pub}, P)e(Q_{ID}, P_{pub})^{-c'})$$

Expand and simplify:

$$c' = H(m, e(cS_{ID}, P)e(rP_{pub} + aP_{pub}, P)e(Q_{ID}, P_{pub})^{-c'})$$

$$c' = H(m, e(S_{ID}, P)^c e((r + a)P, P_{pub})e(Q_{ID}, P_{pub})^{-c'})$$

$$c' = H(m, e(S_{ID}, P)^c e(P_{pub}, (r + a)P)e(Q_{ID}, P_{pub})^{-c'})$$

$$c' = H(m, e(Q_{ID}, P_{pub})^c e(P_{pub}, (r + a)P)e(Q_{ID}, P_{pub})^{-c'})$$

Simplify further using the properties of bilinear pairings:

$$c' = H(m, e(Q_{ID}, P_{pub})^{c-c'})$$

$$c' = H(m, e(Q_{ID}, P_{pub})^b e(R + aP, P_{pub}))$$

$$c' = H(m, e(bQ_{ID} + R + aP, P_{pub}))$$

$$c' = H(m, t)$$

Given that $c = H(m, t) + b \mod q$, we have $c' = c - b$
Thus, the verification equation holds, ensuring the correctness of the protocol.

4.2 scheme 2 [2]

Setup: PKG selects randomly $s \in Z_q$ and computes public key $P_{Pub} = sP$. Publishes PARAM = G1, q, e, P, P_{Pub}, H_1, H_2, H_3, and keep secret key s secretly.

Extract:PKG computes $S_{IDS} = sQ_{IDS}$, and $S_{IDU} = sQ_{IDU}$, where $Q_{IDS} = H_1(IDS)$ and $Q_{IDU} = H_1(IDU)$, and sends S_{IDS} and S_{IDU} to the signer and the user respectively.

Commitment: On randomly chosen integer $r \in Z_q$, the signer computes $k = e(S_{IDS}, rH_2(t)Q_{IDU})$ and $R = rH_2(t)Q_{IDS}$, and passes R and k as a commitment to the user.

Authenticating and Blinding: Using his private key, the user computes $K = e(S_{IDU}, R)$. If kK, the user picks a random number $a \in Z_q$ as a blinding factor, computes $A = a^{-1}R$, $h = H_3(m, A)$, $b_M = ah$ and $X = H_3(b_M, K)$, and sends (b_M, X) to the signer.

Signing:The signer computes $X' = H_3(b_M, k)$ and check if $X' == X$ holds. For valid justification, the signer produces a signature with his private key as $Sig = (rH_2(t) + b_M)S_{IDS}$ and sends it back to the user.

Unblinding: The user unblinds the blinded signature S with blinding factor a as $Sig' = a^{-1}Sig$, and publishes signature Sig', A, m on the message m.

Verify: $e(Sig', P)? = e(A + H_3(m, A)Q_{IDS}, P_{pub})$

Correctness:

$$Sig = (rH_2(t) + aH_3(m, A))SQ_{IDS}$$

$$Sig = (rH_2(t)Q_ids + aH_3(m, A)Q_{IDS})S$$

$$Sig = (R + aH_3(m, A)Q_{IDS})S$$

$$Sig' = Sig.a^{-1}$$

$$Sig' = (A + H_3(m, A)Q_{IDS})S$$

$$e(Sig', P) = e((A + H_3(m, A)Q_{IDS})S, P) = (A + H_3(m, A)Q_{IDS}, SP)$$

4.3 Comparison

now we compare the 2 mentioned schemes with 2 popular blind signatures which are not based on IBE, RSA blind signature, and ECDSA blind signature.

	Key Distribution	Implicit Authentication	Non-forgeability	Token malleability	Security Level
scheme 1	no need for separate public key distribution	yes	yes	secure	GDH
scheme 2	no need for separate public key distribution	yes	yes	secure	GDH
RSA blind signature	needs a CA, more complex	No, requires separate verification mechanisms	yes	Vulnerable	Integer factorization
ECDSA blind signature	needs a CA, more complex	No, requires separate verification mechanisms	yes	secure	Elliptic curve discrete logarithm problem

This section compares scheme 2 with three existing ID-BS schemes. Assuming the pairing operation on elliptic curve is a very time-consuming operation, the Table below shows that scheme 2 needs fewer operations and is much more efficient than [7], [1], [8] schemes.

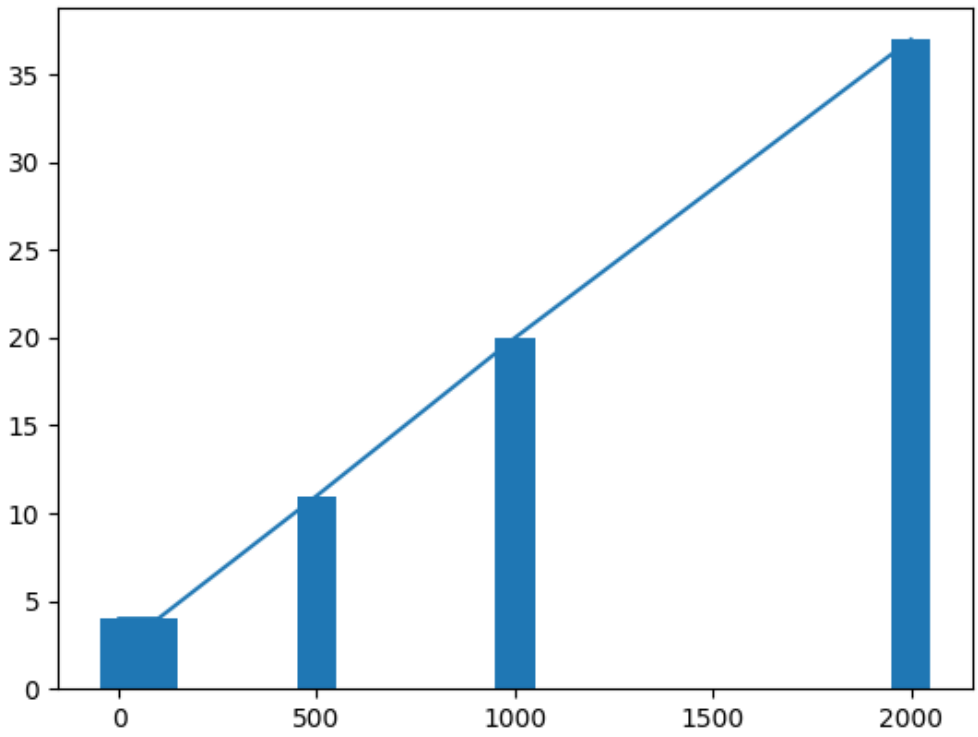
Schemes	pairing operation	mult scaler and G1	addition 2 elements G1	hash function H	two scalar mult	pairing elements compare	exp of pairing	mult two pairing
Zhang and K. Kim 2002	3	6	4	2	0	0	1	1
Z. Huang, K. Chen	6	2	1	2	2	1	4	3
Zhang and K. Kim 2003	2	6	2	2	2	1	0	0
scheme 2	2	5	1	5	2	1	0	0

4.4 Test Result

In this section, we have analyzed the response time of the both schemes for 2000 voters. we can see that the scheme2 is efficient even for high population of the voters.

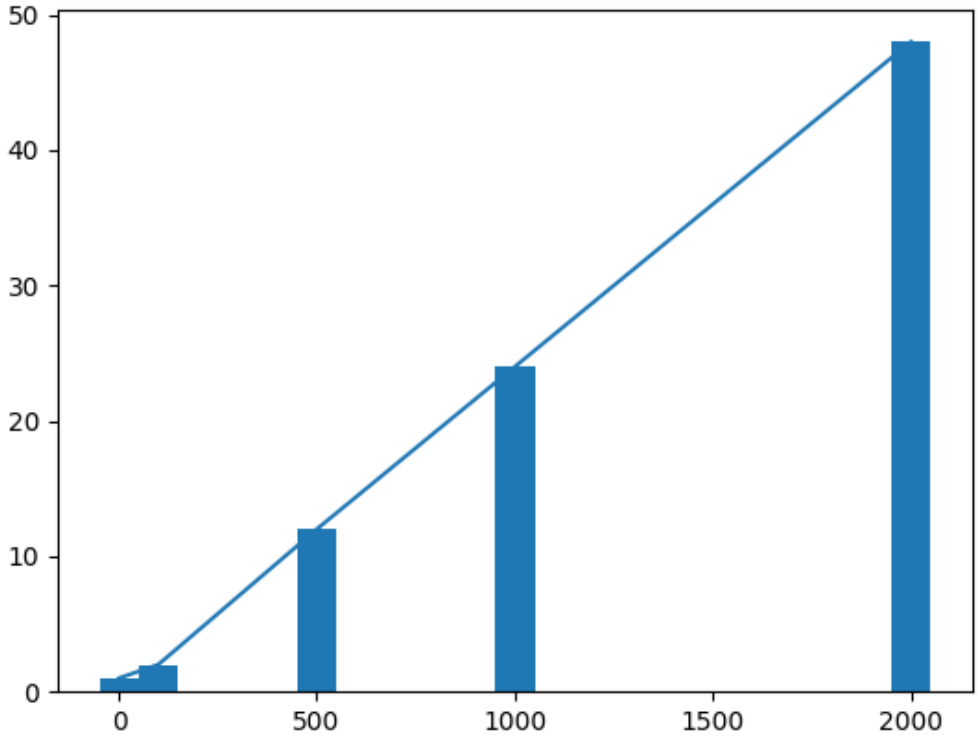
It will take about 30 seconds to finish the signing and verfying for more than 2000 users.

The X-axis is the number of voters and the Y-axis is the time taken in seconds.

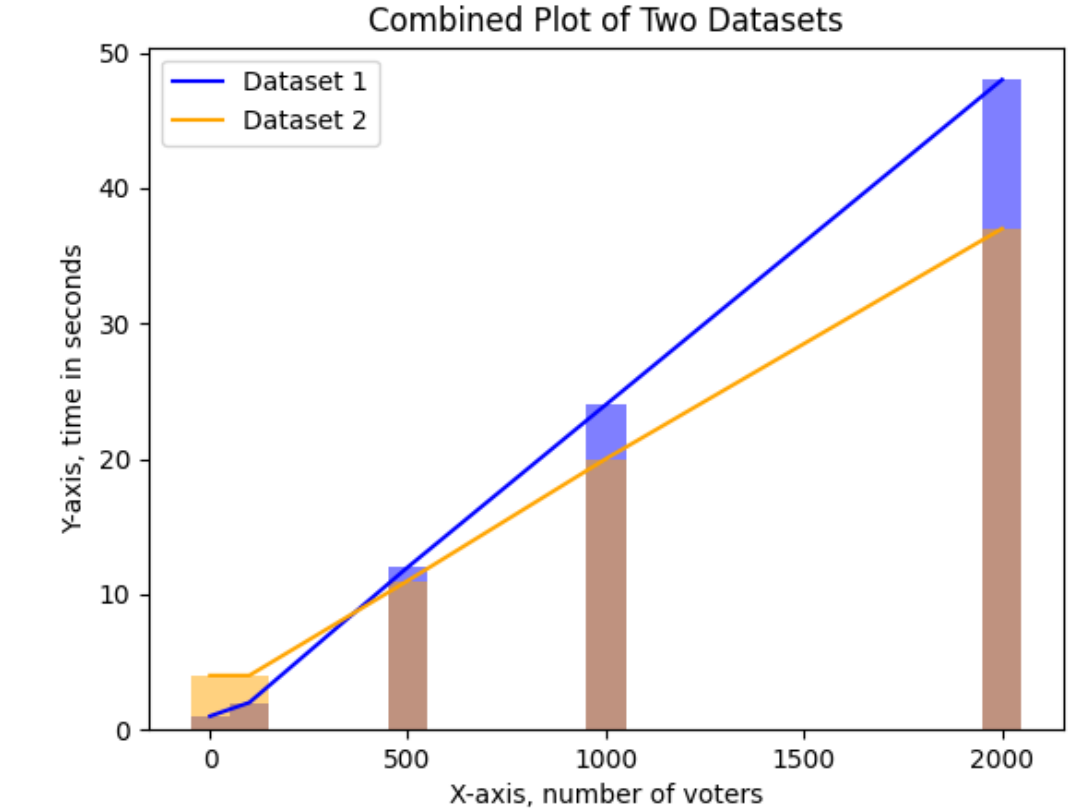


---

for scheme1, we found out that although the response time is faster for a few users, but for high population, scheme2 is working better. The below plot is the response time for signing and verifying the scheme1. The X-axis is the number of voters and the Y-axis is the time taken in seconds.



And here is how efficient the shceme2 will be for more users:



5 Electronic voting with IBE [2]

5.1 Overview

E-Voting using Blind Signatures Blind signatures are crucial in e-voting systems as they enable voter privacy while ensuring the authenticity of votes. By allowing voters to obtain a signature on their ballot without revealing its content, blind signatures prevent the linking of voters to their choices, thus maintaining anonymity. Additionally, they ensure that each vote is valid and has been issued by an authorized entity, enhancing the overall security and integrity of the election process.

5.2 Mahender Kumar E-voting System Design [2]

this system is based on the identity-based blind signature scheme 2, which we described before, and consists of 5 parties:
Voter, Authentication Party(AP), Vote Casting Party(VCP), Vote Tallying Party(VTP), Trusted Third Party(TTP).

now we explain 4 algorithms:

5.2.1 Registration

TTP selects random integer $s \in Z_q$ and computes the public key $P_{pub} = sP$ and then publishes PARAM=$P, sP, G, q, H_1, H_2, H_3$ and keeps s secretly. AP and voter registered themselves against their Identity ID_A and ID_V.Using his master key, s, TTP computes private keys $SID_A = sQ_IDA$

and $SID_V = sQ_{ID_V}$ where $Q_{ID_A} = H_1(ID_A)$ and $Q_{ID_V} = H_1(ID_V)$, and sends SID_A and SID_V to the AP and the voter.

5.2.2 Authentication

The AP chooses a secret random integer $r \in Z_q$, computes k and R where, $k = e(SID_A, rH_2(t)Q_{ID_V})$ and $R = rH_2(t)QID_A$, and delivers R to the voter.

Using private key SID_V, the voter computes $K = e(SID_V, R)$. If any forger wants to compute k with his private key SID_f, he can't compute the next step correctly because of kK. Only an authenticated voter can proceed. Now, Voter chooses a random number $a \in Z_q$ as blinding factor, computes $A = a^{-1}R$, h, b_M and X, where, $h = H3(m, A)$, $b_M = ah$ and $X = H_3(b_M, K)$. Now, the voter sends b_M and X to the AP.

On given blinded message (b_M, X), AP computes $X' = H_3(b_M, k)$. AP signs the blinded message with his private key as $S' = b_M SID_S$ if and only if X' and X are equals.

Upon receiving the blinded signature S from AP, the voter strips it to compute the signature as $S' = a^{-1}S$.

5.2.3 Vote casting

the voter computes $V = aH_1(vote)$, where vote $\in 0, 1^*$ and electronic ballot B and send to VCU, where B = S', A, R, vote, V. Then, VCP checks two-phase verification as

$$e(S, P)? = e(hQ_{ID_A}, P_{pub})$$

$$e(V, A)? = e(H_1(vote), R)$$

Identical to verify the algorithm of the ID-BS scheme, for every valid verification the VCP keeps it as a valid ballot, otherwise, it leaves it as an invalid. Now, VCP selects a random $x \in Z_q$ and generates a receipt Rcpt and signature on receipt S_{Rcpt} with his private key SID_C to prevent vote coercion, where:

$$Rcpt = H_1(B||x)$$

$$S_{Rcpt} = H_2(Rcpt)SID_C$$

VCP sends the receipt and signature $(Rcpt, S_{Rcpt})$ to the voter.
The voter checks $S_{Rcpt}, QID_V, RcptQ_{ID_C}, SID_V$ is the valid tuple of GDP and verifies if the following equation holds:

$$e(S_{Rcpt}, Q_{ID_V})? = e(RcptQ_{ID_C}, SID_V)$$

5.2.4 Vote counting

After voting, VTP completes the vote counting and then filters identical or electronic fraud ballots. Suppose $B_i = S', A_i, R_i, vote_i, V_i$ and $Bi + 1 = Si + 1', Ai + 1, Ri + 1, vote_{i+1}, V_{i+1}$ are two ballots in the electronic ballot list. The signatures S_i' on A_i and Signature V_i and on $vote_i$ are generated using the two randomly chosen integers a_i so the signatures must be unique. The VTP filters the invalid voter by comparing the two ballots with their signature. If $S_i' == S_{i+1}'$ and $V_i == V_{i+1}$ in the stored list of electronic ballots are the same, one ballot is considered invalid and the other is valid. The VTP considered the first ballot as valid and invalidated the vote. To count the valid votes, the VTP

maintains two lists: the first list contains the valid ballots with their corresponding receipt *Rcpt*, and the other list includes all invalid ballots with their receipts *Rcpt*. then publishes the two lists. (list of pairs of *Rcpt*s with the public A in the ballot)

5.2.5 Security Analyse

- Authentication: In the authentication phase, AP authenticates the voter, signs on the blinded ballot, and sends a blindly signed ballot to the voter.
- Coercion-resistant: The property of coercion-resistant is provided by the use of the random number x which gives the randomness in Rcpt.
- Verifiability: After the vote counting stage, VTP published two lists containing the names of valid and invalid voters with their Rcpt.
- Integrity: In the vote-casting phase, the voter signs the vote with his randomly chosen number a, i.e., $V = aH_1(vote)$. If any entity wants to alter the value of a vote, he must have to guess the exact value of a, which is equivalent to solving the ECDLP problem.

5.2.6 Security Challenges

Universal Verifiability: VTP publishes 2 lists consisting of *Rcpt* and ID, pairs and according to the fact that all used hash functions are pre-image resistance, thus no one can find the exact vote using the *Rcpt*. thus no one can verify that all votes are counted correctly and each voter can just verify that his vote is counted or not.

5.3 Our Contribution

to solve the problem of universal verifiability we can change the design as follows, denote that the authentication algorithm is as same as before:

5.3.1 new Vote casting

in this algorithm the voter has 3 votes, v_1, v_2, v_3. the correct format for votes is: 2 of these votes should be the same and the other one should be different. otherwise, the VTP would not accept the vote.
the voter computes $V = aH_1(v_1, v_2, v_3)$, where votes are $\in 0, 1^*$ and electronic ballot B and send to VCU, where B = S', A, R, list of votes, V. Then, VCP checks two-phase verification as

$$e(S, P)? = e(hQ_{ID_A}, P_{pub})$$

$$e(V, A)? = e(H_1(list\ of\ votes), R)$$

Identical to verify the algorithm of the ID-BS scheme, for every valid verification the VCP keeps it as a valid ballot, otherwise, it leaves it as an invalid. Now, VCP selects a random $x \in Z_q$ and generates 3 receipts $Rcpt_1$,$Rcpt_2$,$Rcpt_3$ as follows:

$$Rcpt_i = H_1(v_i||x_i)$$

then it asks the voter to choose an $i \in 1, 2, 3$ and then sends $< Rcpt_i, x, v_i >$ and the signature with his private key SID_C to the voter.

$$S_{Rcpt_i} = H_2(Rcpt_i)SID_C$$

The voter checks $S_{Rcpt}, QID_V, RcptQ_{ID_C}, SID_V$ is the valid tuple of GDP and verifies if the following equation holds:

$$e(S_{Rcpt}, Q_{ID_V})? = e(RcptQ_{ID_C}, SID_V)$$

also, he checks:

$$Rcpt = H_1(v_i||x_i)$$

5.3.2 new Vote counting

consider x is the number of votes on 0 and y is the number of votes on 1, suppose that x>y. the VTP computes the number of voters who voted on $0(nv_0)$ and $1(nv_1)$ as follows:

$$z = x - y = nv_0 - nv_1$$

$$nv_0 + nv_1 = N \ \rightarrow \ the\ number\ of\ all\ voters$$
$$\rightarrow \frac{N - z}{2} = nv_0$$
$$\rightarrow \frac{N + z}{2} = nv_1$$

then he publishes a list of pairs of *Rcpt* and votes and also publishes a separate list of voters.

5.3.3 Conclusion

in the new system design, we have supposed a way to provide universal verifiability, so each voter can compare his *Rcpt* with the published one, check the correctness of that, and check the sum of all votes. In this scheme no one can prove his vote to another one because each voter has just one of his 3 *Rcpt*'s thus it is Coercion resistance.

5.4 Comparison

in this section, we compare our proposed enhanced system with the ThreeBallot Rivest [3] system and Mahender Kumar's system [2].

e-voting system design	Eligibility	Coercion resistance	Universal Verifiability	Voter anonymity
Mahender Kumar	yes	yes	no	yes
our propossed scheme	yes	yes	yes	yes
ThreeBallot Rivest	no	yes	yes	no

6 Conclusion

In this report, we first provided a formal definition of Identity-Based Encryption (IBE) and then described the formal definition and usage of Blind Signatures. Following this, we introduced two Identity-Based Blind Signature schemes. As a practical application of Blind Signatures, we described an e-voting system that utilizes Blind Signatures for its authentication phase. Additionally, we proposed our idea for improving the mentioned system. We implemented and analyzed the signatures and compared them with some other schemes. This comparison demonstrated that our schemes offer significant improvements in terms of security and efficiency. These enhancements make them suitable options for e-voting systems and other similar applications. Furthermore, the results from our analysis and implementation indicate that the proposed schemes can effectively manage various security threats.

References

[1] Zhenjie Huang, Kefei Chen, and Yumin Wang. Efficient identity-based signatures and blind signatures. pages 120–133, 12 2005.

[2] Mahender Kumar, C.P. Katti, and P. Saxena. *A Secure Anonymous E-Voting System Using Identity-Based Blind Signature Scheme*, pages 29–49. 01 2017.

[3] Ronald Rivest. The threeballot voting system. 11 2006.

[4] Luke Valenta and Brendan Rowan. Blindcoin: Blinded, accountable mixes for bitcoin. pages 112–126, 01 2015.

[5] Xun Yi and Kwok-Yan Lam. A new blind ecdsa scheme for bitcoin transaction anonymity. pages 613–620, 07 2019.

[6] Fangguo Zhang and Kwangjo Kim. Id-based blind signature and ring signature from pairings. volume 2501, 12 2002.

[7] Fangguo Zhang and Kwangjo Kim. Id-based blind signature and ring signature from pairings. volume 2501, 12 2002.

[8] Fangguo Zhang and Kwangjo Kim. Efficient id-based blind signature and proxy signature from bilinear pairings. pages 312–323, 07 2003.