A Tamper-Evident Voting Machine Resistant to Covert Channels

Han Wei[1] Hao Tao[1] Zheng Dong[1] Chen Ke-fei [1] Chen Xiaofeng [2]

(1. Shanghai Jiaotong University, Shanghai, 200240, P.R.China)

(2. Sun Yat-sen University, Guangzhou 510275, P.R.China)

Abstract:

To provide a high level of security guarantee cryptography is introduced into the design of the voting machine. The voting machine based on cryptography is vulnerable to attacks through covert channels. An adversary may inject malicious codes into the voting machine and make it leak vote information unnoticeably by exploiting the randomness used in encryptions and zero-knowledge proofs. In this paper a voting machine resistant to covert channels is designed. It has the following properties: Firstly, it is tamper-evident. The randomness used by the voting machine is generated by the election authority. The inconsistent use of the randomness can be detected by the voter from examining a destroyable verification code. Even if malicious codes are run in the voting machine attacks through subliminal channels are thwarted. Next, it is voter-verifiable. The voter has the ability to verify if the ballot cast by the machine is consistent with her intent without doing complicated cryptographic computation. Finally, the voting system is receipt-free. Vote-buying and coercion are prevented.

Keywords: electronic voting, covert channel, tamper-evident, receipt-free

1. Introduction

Electronic voting will change the way we vote in the near future. Direct Recording Electronic (DRE) voting machines have been widely used in many political elections in recent years. The inner mechanism of the DRE is opaque to voters, which results in the debate on the trustworthiness of the machine. One of the most challenging issues is to design voting systems that can be trusted by human voters even if the election computers are running malicious codes. There have been some voting schemes in which a voter can get direct verification that her vote is correctly recorded by the DRE, such as Neff's MarkPledge scheme [1] and Moran-Naor's scheme [2]. The two schemes are based on cryptography and they are vulnerable to attacks through covert channels. A corrupt programmer may write malicious codes to leak voters' choices by encoding the randomness used in encryptions or zero-knowledge proofs in a secret and determinant way. When she has access to the bulletin board she can deduce the content of an encrypted ballot by reading publicly available information. We take Moran-Noar's voting scheme as an example. A voter operates the DRE in the voting booth as follows:

1. The voter chooses one candidate whom she supports.
2. The voter inputs challenges for other candidates.

3. The voter waits till the DRE prints a commitment to her choice on the receipt.

4. The voter inputs the challenge for the candidate she chooses.

5. The voter waits till the DRE prints the rest of the receipt. She compares the challenges in her mind with the challenges printed on the receipt. If they are consistent she takes the receipt and leaves the voting booth, or else she complains to the voting authority.

In such a scheme attacks can be mounted through covert channels. We divide the adversaries into two types: weak adversary and strong adversary. The weak adversary does not interact with the voter. The weak adversary only injects malicious codes into the DRE. When the DRE uses randomness it encodes the voter's choice into the random string in a private way which is known to the adversary. A strong adversary may be able to both inject malicious programs into the DRE and coerce voters. A strong adversary can mount attacks in the following cases:

Case 1. The adversary can require a coerced voter to use a special challenge which is recognized by the DRE. Once it knows a voter is coerced, the DRE can change the vote as it wishes. The coerced voter will not complain even if she detects the inconsistence.

Case 2. The adversary can control the order of voters entering the voting booth. The voting order is stored into the DRE. So the DRE can distinguish the coerced voters and change their ballots.

Case 3. The adversary can make use of timing channels. The time interval of striking keys on the keyboard of the DRE can be used to identify a coerced voter.

In this paper, we make the assumption that the arrival of voters is random so we can ignore case 2. We leave case 3 as an open problem. We only consider how to defend against the weak adversary and the strong adversary of case 1.

The attacks through covert channels are cumbersome. Public code audit may mitigate the effect to some extent, however, it is difficult to ascertain that the audited code is the actual code that gets loaded into the DRE. In order to escape detection a corrupt software developer may instruct the DRE to switch its behavior to a correct mode in the course of code test.

To prevent covert channels we can require the randomness used by the DRE is generated by the voting authority. The voting authority may be composed of several election trustees who generate the randomness via multi-party computation. In such a way trust is distributed. The construction of witness of correct randomness use is difficult since it must be done without exposing the randomness used, and it must not introduce new covert channels. Choi, Golle and Jakobsson presented the design of tamper-evident mix networks with auditable privacy [3]. Inspired by their ideas, a tamper-evident DRE voting machine is designed in this paper. The voter acts as a verifier who checks that the randomness used by the DRE is correctly generated by the voting authority.

The rest of the paper is organized as follows: the voting system model is outlined in section 2. Some building blocks are presented in section 3. The voting scheme is detailed in section 4. Its security is analyzed in section 5 and the conclusion is drawn in section 6.

2. System model

In this paper we consider a 1-out-of-L voting scheme in which a voter chooses a candidate out of L candidates. Voters cast their ballots in a private voting booth. The following entities are involved: the voter, the DRE, the voting authority, the adversary and the assistant verifier.

The voter enters the voting booth and casts a ballot on the DRE. She audits that all the randomness used by the DRE is generated by the voting authority and that the DRE casts a ballot as intended.

The DRE generates encrypted ballots and posts them to a public bulletin board. The DRE prints a cryptographic receipt for the voter.

The voting authority maintains a list of all eligible voters. The authority is responsible to generate the randomness used by the DRE in encryptions. The authority sends each voter a verification code out of band, which is used by the voter to check if the given randomness is used by the DRE.

The adversary tries to mount attacks through covert channels. It is allowed to inject malicious codes into the DRE.

The assistant verifier is able to execute cryptographic computation. It can be any third party the voter trusts. When the voter leaves the booth, she gives her receipt to the verifier. The assistant verifier checks if the receipt is generated in accordance with some pre-defined rules.

Next we consider communication channels in the voting scheme: The bulletin board model is widely employed in electronic voting schemes. It is a public broadcast channel. Only eligible users can append messages on it. Messages posted cannot be tampered with. In our voting scheme the bulletin board is used for the authority and the DRE to publish information related to voting. The voting booth is modeled as an untappable channel between the voter and the DRE. The adversary can communicates with the DRE before it is deployed at the polling station. From this moment on, the adversary has no access to the DRE, and except the bulletin board the DRE has no communication media to transmit information to the adversary.

3. Preliminaries

Security Requirements. A voting protocol should satisfy the security requirements below:

Eligibility: Only eligible voters can participate in the election, and each eligible voter can cast a single vote.

Privacy: The content of an individual ballot is kept secret. Only the final tally result is published.

Verifiability: The validity of the individual ballot and the tally process can be verified. When any passive third party can also verify the ballot cast and the tally, this property is called universal verifiability.

Robustness: The voting protocol can tolerate corrupt voters and dishonest authorities to some extent.

Fairness: The partial results of the tally should not be exposed prior to the end of the voting phase.

Receipt-freeness: The voter cannot provide a receipt to convince others how she casts a ballot. If a voter is given a cryptographic receipt for verification and the receipt can not be used to convince others that the voter chooses a particular candidate, the voting protocol still satisfies the receipt-freeness.

Encoding votes. We denote by M a strict upper bound on the number of voters. We represent L candidates with numbers $0,...,L-1$ and encode a vote on candidate i as M^i. Tallying such encoded votes gives us an M-addic representation of the result $\sum_{i=0}^{L-1} v_i M^i$, where v_i is the number of votes on candidate i .

Homomorphic encryption. In this paper we make use of a semantically secure homomorphic threshold cryptosystem. A probabilistic public-key encryption function $E: P \times R \to C$ is homomorphic if for all $x_1, x_2 \in P$, $r_1, r_2 \in R$, it holds that $E(x_1; r_1) \boxtimes E(x_2; r_2) = E(x_1 + x_2; r_1 \oplus r_2)$, where P is a group which is the plaintext space, R is a group which is the randomness space, and C is a group which is the ciphertext space. The group operations in P, in R and in C are denoted by "$+$", "$\oplus$" and "$\boxtimes$" respectively. Examples of homomorphic cryptosystems are the additive version [4] of ElGamal [5] and Paillier [6]. They both are semantically secure and have threshold variants ([7] [8] [9]).

A cut-and-choose zero-knowledge proof protocol. Suppose E is a homomorphic encryption function. A prover P presents a verifier V with a ciphertext $c = E(m; r)$ and claims that it encrypts the plaintext m . By the homomorphic property P can convince V without disclosing r by the protocol below:

Common input: c

Private input for P: r such that $c = E(m; r)$

1. P chooses $r_0, r_1 \in R$ at random such that $r = r_0 \oplus r_1$. P computes $c_0 = E(m/2; r_0)$, $c_1 = c \boxed{\div} c_0$. "$\boxed{\div}$" is the inverse operation of "$\boxtimes$". P sends c_0, c_1 to V .
2. V chooses $b \in \{0,1\}$ randomly and sends b to P .
3. P sends r_0 to V if $b = 0$, sends r_1 to V if $b = 1$. V checks that $c = c_0 \boxtimes c_1$ and $c_b = E(m/2; r_b)$.

The protocol is a typical construction with soundness 1/2. It can be repeated k times to make the error probability achieve $1/2^k$.

4. The voting scheme

Step 1. Voting system initialization

The voting authority posts the list of eligible voters on the bulletin board. It generates the key pairs of a homomorphic encryption cryptosystem. The public key and a cryptographic hash function "*hash*" are published on the bulletin board and loaded into the firmware of the DRE. The secret key may be distributed in a threshold shared manner. The voting authority deploys DREs into polling stations.

The voting authority generates ballots for each voter as follows: ① It picks $r_1, r_2, ..., r_L \in R$ randomly and generates a *ballot set* $BS = \{ E(A_1; r_1),\ E(A_2; r_2),\ ...,\ E(A_L; r_L) \}$ including encrypted valid votes. We denote by A_i the coding of candidate i $(i = 1, 2, ..., L)$ for notation convenience. $E(A_i;r_i)$ is a valid encrypted vote for candidate i. ② It generates a unique id for the ballot set BS. ③ For each encrypted ballot $E(A_i;r_i)$, the authority picks k pairs of random numbers $(r_{i,0}^{(1)},r_{i,1}^{(1)})$, $(r_{i,0}^{(2)},r_{i,1}^{(2)})$, ..., $(r_{i,0}^{(k)},r_{i,1}^{(k)})$ in R such that $r_{i,0}^{(s)}+r_{i,1}^{(s)}=r_i$ $(s=1,2,...,k)$. Here k is a security parameter. The authority computes the row vector [($E(A_i/2;r_{i,0}^{(1)})$, $E(A_i/2;r_{i,1}^{(1)})$), ($E(A_i/2;r_{i,0}^{(2)})$, $E(A_i/2;r_{i,1}^{(2)})$), ..., ($E(A_i/2;r_{i,0}^{(k)})$, $E(A_i/2;r_{i,1}^{(k)})$)]. In the row vector each element is a pair of encryptions on $A_i/2$ and the equation $E(A_i/2;r_{i,0}^{(s)}) \boxtimes E(A_i/2;r_{i,1}^{(s)}) = E(A_i;r_i)$ holds. All of the row vectors forms a *ballot set matrix*

$$BSM=\begin{bmatrix} (E(A_1/2;r_{1,0}^{(1)}),E(A_1/2;r_{1,1}^{(1)})) & (E(A_1/2;r_{1,0}^{(2)}),E(A_1/2;r_{1,1}^{(2)})) & \dots & (E(A_1/2;r_{1,0}^{(k)}),E(A_1/2;r_{1,1}^{(k)})) \\ (E(A_2/2;r_{2,0}^{(1)}),E(A_2/2;r_{2,1}^{(1)})) & (E(A_2/2;r_{2,0}^{(2)}),E(A_2/2;r_{2,1}^{(2)})) & \dots & (E(A_2/2;r_{2,0}^{(k)}),E(A_2/2;r_{2,1}^{(k)})) \\ \dots & \dots & \dots & \dots \\ (E(A_L/2;r_{L,0}^{(1)}),E(A_L/2;r_{L,1}^{(1)})) & (E(A_L/2;r_{L,0}^{(2)}),E(A_L/2;r_{L,1}^{(2)})) & \dots & (E(A_L/2;r_{L,0}^{(k)}),E(A_L/2;r_{L,1}^{(k)})) \end{bmatrix}_{L\times k}$$

. ④ The authority computes the hash values of the matrix BSM and presents the hash values in the form of a matrix

$$HBSM=\begin{bmatrix} (Hash(E(A_1/2;r_{1,0}^{(1)})),Hash(E(A_1/2;r_{1,1}^{(1)}))) & \dots & (hahs(E(A_1/2;r_{1,0}^{(k)})),hash(E(A_1/2;r_{1,1}^{(k)}))) \\ \dots & \dots & \dots \\ (hash(E(A_L/2;r_{L,0}^{(1)})),hash(E(A_L/2;r_{L,1}^{(1)}))) & \dots & (hash(E(A_L/2;r_{L,0}^{(k)})),hash(E(A_L/2;r_{L,1}^{(k)}))) \end{bmatrix}_{L\times k}$$

. The matrix $HBSM$ is the verification code for the DRE. Each row corresponds to the encrypted vote for a candidate. The authority prints the verification code on a sheet of paper and inserts the sheet into an envelope. The envelope is sealed and the ballot id number is printed on the surface of the envelope.

Step 2. Casting a ballot

On the election day the authority loads the ballot information including the id of the ballot set and all of the randomness used in the ballot generation into the DRE. The sealed envelopes are transported to each polling station.

After a voter's identity is verified, she is given a sealed envelope. The verification code is inside the envelope. The voter generates an k-bit string in a random manner such as flipping a coin under the inspection of the polling workers. The voter publishes the string as her challenge which will be used in the voting booth.

The voter enters the voting booth and starts the DRE. The id number shown on the screen should be equal to the id printed on the envelope. Assume that the voter chooses the candidate t $(1\le t\le L)$ on the DRE. The DRE chooses the encryption $E(A_t;r_t)$ from the given ballot set *BS*. The DRE prints $E(A_t;r_t)$ on the receipt and posts it on the bulletin board. After the encryption is printed, the voter inputs the k-bit challenge string into the DRE. For the l-th bit b_l

$(l=1,2,...,k)$, if $b_l=0$, the DRE reveals $r_{i,0}^{(l)}$ in the l-th column of the matrix BSM; if $b_l=1$, the DRE reveals $r_{i,1}^{(l)}$ in the l-th column of the matrix BSM $(i=1,2,...,L)$. The revealed randomness is printed on the receipt. The DRE computes a matrix BSM' as follows: first, let $BSM=BSM'$. Next, some elements in BSM' will be modified. The t-th row in matrix BSM' is not changed. For the i-th row $(i \neq t)$, in each encryption pair ($E(A_i/2;r_{i,0}^{(s)})$, $E(A_i/2;r_{i,1}^{(s)})$) $(s=1,2,...,k)$ the revealed encryptions $E(A_i/2;r_{i,b_s}^{(s)})$ are maintained. The unrevealed encryptions $E(A_i/2;r_{i,\bar{b}_s}^{(s)})$ are modified as $E(A_t;r_t) \boxminus E(A_i/2;r_{i,b_s}^{(s)})$.
The DRE computes the hash value of each element in matrix BSM' to obtain a matrix $HBSM'$. The challenge and the matrix $HBSM'$ are printed on the receipt.

Step 3. Verification inside the voting booth

The voter opens the envelope and takes out the verification code. She compares the matrix $HBSM$ with the matrix $HBSM'$. The t-th row in $HBSM$ should be the same as that in $HBSM'$. In other rows, in each pair of hash values if $b_l=0$ the two left components should both be equal to $hash(E(A_i/2;r_{i,0}^{(l)}))$, otherwise the two right components should both be equal to $hash(E(A_i/2;r_{i,1}^{(l)}))$ $(l=1,2,...,k)$. If the verification fails the voter must immediately complain to the polling workers.

Step 4. Verification outside the voting booth

The voter takes the receipt and goes out of the booth. She destroys the verification code under the inspection of the polling workers. Polling workers check that the challenge string printed on the receipt is the same as the one the voter committed to before entering the booth. If the two challenge strings are inconsistent, the ballot cast by the voter is cancelled and the cancellation is announced on the bulletin board at once. The voter submits her receipt to the assistant verifier. The verifier checks: ① The encrypted ballot $E(A_t;r_t)$ on the receipt has been published on the bulletin board. ②The matrix $HBSM'$ is correctly computed by the hash function. Note that the verifier can use the revealed randomness $r_{i,b_l}^{(l)}$ to reconstruct the matrix BSM'. ③ In the matrix BSM' $E(A_t;r_t)$ equals the product of two components in each pair. If any inconsistence is detected, the assistant verifier will raise an alert and complain to the legal authority on behalf of the voter.

Step 5. Tallying

When the voting phase ends, the voting authority can aggregate the ciphertexts of the ballots by the homomorphic property of the cryptosystem. The aggregated ciphertext may be threshold decrypted by multiple authorities. In this case a zero-knowledge proof is posted on the bulletin board to convince the public that the decryption is correctly performed. It is straightforward to extract the voting result from the plaintext. Using tamper-evident mix networks [3] to tally is an alternative choice.

Step 6. Verifying the tally

Any passive third party interested in the election can re-compute the aggregated ciphertext and verify the zero-knowledge proof of correct decryption.

5. Security analysis and discussion

The proposed voting scheme satisfies the requirements of electronic voting:

Eligibility. The list of eligible voters is published on the bulletin board. The voter's identity is verified before she enters the voting booth.

Privacy. The ballot posted on the bulletin board is encrypted. The verification code is destroyed when the voter leaves the booth. The receipt does not reveal the content of the encrypted ballot.

Verifiability. The tally can be re-computed and verified by any third party by using information published on the bulletin board.

Robustness. The encryptions of ballots are generated by the voting authority. If the DRE fails to encrypt a ballot, the voter can detect the mistake by comparing the verification code with the receipt.

Fairness. No single encrypted ballot is decrypted. A voter makes her choice at her disposal.

Receipt-freeness. The encrypted ballot is not generated by the voter. If the verification code and the receipt can be seen at the same time, the voter's choice can be deduced. But the verification code is destroyed. Just by reading the receipt we can see the probability for each candidate is equal. So the scheme is receipt-free.

Next we will show the voting scheme is secure against covert channels. The encrypted ballot set is generated by the authority. All the computation performed by the DRE is deterministic. There is no way for the DRE to encode secret information into the randomness. The voter is forced to generate her challenge string in a random manner just before she enters the booth. The DRE cannot distinguish the coerced voters from average voters by recognizing the challenges.

The voting scheme is voter verifiable. The voter can check that the DRE casts a ballot as intended and that the voting authority generates encrypted ballots correctly. If the encrypted ballot is inconsistent with the voter's choice, the DRE will fail to open the randomness in the matrix *BSM* with the probability $1-1/2^k$. It is easy for the voter to detect the mistake.

In real life thousands of DREs may be deployed at polling stations in a political election, and DREs may be manufactured by different vendors. It is hard to examine all the voting machines thoroughly. In the proposed voting scheme the generation of the encrypted ballots is centralized so the process is ease to monitor. The DRE is inspected by the voter. The overhead of malicious code test is decreased.

6. Conclusion

In this paper an electronic voting scheme is presented. Especially, the DRE is designed to achieve tamper-evidence and the attacks from covert channels are prevented. The voter verifies that the DRE uses pre-defined randomness by examining a destroyable verification code. Although the voting scheme is based on cryptography, it is unnecessary for the voter to do cryptographic computation. Only the operation of string comparison is performed by the voter. The complicated cryptographic computation can be carried out by any third party outside the booth. Although a voter is given a receipt to verify if the ballot is cast as intended, the receipt cannot be used to convince others which candidate the voter chooses. The vote-buying and coercion are thwarted.

References

[1] C. A. Neff, "Practical high certainty intent verification for encrypted votes", http://votehere.com/old/vhti/documentation/vsv-2.0.3638.pdf, 2004.

[2] T. Moran and M. Naor, "Receipt-free universally-verifiable voting with everlasting privacy", CRYPTO 2006, LNCS vol.4117, Springer-Verlag, pp. 373-392, 2006.

[3] J. Y. Choi, P. Golle and M. Jakobsson, "Auditable privacy: on tamper-evident mix networks", Financial Cryptography 2006, LNCS vol. 4107, Springer-Verlag, pp.126-141, 2006.

[4] R. Cramer, R. Gennaro and B. Schoenmakers, "A Secure and Optimally Efficient Multi-Authourity Election Scheme", EUROCRYPT 1997, LNCS Vol. 1233, Springer-Verlag, pp. 103-118, 1997.

[5] T. ElGamal, "A Public key cryptosystem and a signature scheme based on discrete logarithms", CRYPTO'84, LNCS Vol. 196, Springer-Verlag, pp. 10-18, 1984.

[6] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Class", EUROCRYPT 1999, LNCS Vol.1592, Springer-Verlag, pp. 223-239, 1999.

[7] T. P. Pedersen, "A threshold cryptosystem without a trusted third party", EUROCRYPT 1991, LNCS Vol.547, Springer-Verlag, pp. 522-526, 1991.

[8] P. A. Fouque, G. Poupard and J. Stern, "Sharing Decryption in the Context of Voting or Lotteries", Financial Cryptography 2000, LNCS Vol.1962, Springer-Verlag, pp. 90-104, 2001.

[9] I. Damgard and M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-key System", PKC 2001, LNCS Vol. 1992, Springer-Verlag, pp. 119-136, 2001.

Appendix: a concrete example

To explain the voting scheme in detail, we give a simple example here. Assume that there are two candidate and $k=2$. The candidates are encoded as A_1, A_2. The voting authority works as follows:

① The voting authority generates the encrypted ballot set $BS = \{\, E(A_1; r_1)\,, E(A_2; r_2)\,\}$, $r_1, r_2 \in R$.

② The voting authority creates an id for the set *BS*.

③ The voting authority computes the matrix

$$BSM = \begin{bmatrix} (E(A_1/2; r_{1,0}^{(1)}), E(A_1/2; r_{1,1}^{(1)})) & (E(A_1/2; r_{1,0}^{(2)}), E(A_1/2; r_{1,1}^{(2)})) \\ (E(A_2/2; r_{2,0}^{(1)}), E(A_2/2; r_{2,1}^{(1)})) & (E(A_2/2; r_{2,0}^{(2)}), E(A_2/2; r_{2,1}^{(2)})) \end{bmatrix}_{2\times 2}$$

④ The voting authority computes the matrix

$$HBSM=\begin{bmatrix}(Hash(E(A_1/2;r_{1,0}^{(1)})),hash(E(A_1/2;r_{1,1}^{(1)}))) & (hash(E(A_1/2;r_{1,0}^{(2)})),hash(E(A_1/2;r_{1,1}^{(2)})))\\(hash(E(A_2/2;r_{2,0}^{(1)})),hash(E(A_2/2;r_{2,1}^{(1)}))) & (hash(E(A_2/2;r_{2,0}^{(2)})),hash(E(A_2/2;r_{2,1}^{(2)})))\end{bmatrix}_{2\times 2}$$

The matrix *HBSM* is printed on the receipt as the verification code.

Assume the voter chooses the candidate A_2. The DRE prints $E(A_2;r_2)$ on the receipt. If the voter uses the challenge bit string "10", the randomness revealed is $\begin{bmatrix}r_{1,1}^{(1)} & r_{1,0}^{(2)}\\r_{2,1}^{(1)} & r_{2,0}^{(2)}\end{bmatrix}$. The DRE can compute the matrix

$$BSM=\begin{bmatrix}(E(A_1;r_1)\boxed{\div}E(A_1/2;r_{1,1}^{(1)}),E(A_1/2;r_{1,1}^{(1)})) & (E(A_1/2;r_{1,0}^{(2)}),E(A_1;r_1)\boxed{\div}(E(A_1/2;r_{1,0}^{(2)}))\\(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,1}^{(1)}),E(A_2/2;r_{2,1}^{(1)})) & (E(A_2/2;r_{2,0}^{(2)}),E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,0}^{(2)}))\end{bmatrix}_{2\times 2}$$

The DRE modifies some values in *BSM* to obtain the matrix

$$BSM'=\begin{bmatrix}(E(A_2;r_2)\boxed{\div}E(A_1/2;r_{1,1}^{(1)}),E(A_1/2;r_{1,1}^{(1)})) & (E(A_1/2;r_{1,0}^{(2)}),E(A_2;r_2)\boxed{\div}(E(A_1/2;r_{1,0}^{(2)}))\\(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,1}^{(1)}),E(A_2/2;r_{2,1}^{(1)})) & (E(A_2/2;r_{2,0}^{(2)}),E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,0}^{(2)}))\end{bmatrix}_{2\times 2}$$

The DRE computes the matrix

$$HBSM'=\begin{bmatrix}(hash(E(A_2;r_2)\boxed{\div}E(A_1/2;r_{1,1}^{(1)})),hash(E(A_1/2;r_{1,1}^{(1)}))) & (hash(E(A_1/2;r_{1,0}^{(2)})),hash(E(A_2;r_2)\boxed{\div}(E(A_1/2;r_{1,0}^{(2)})))\\(hash(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,1}^{(1)})),hash(E(A_2/2;r_{2,1}^{(1)}))) & (hash(E(A_2/2;r_{2,0}^{(2)})),hash(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,0}^{(2)})))\end{bmatrix}_{2\times 2}$$

The voter compares the matrix *HBSM* with the matrix *HBSM'*:

$$HBSM=\begin{bmatrix}(Hash(E(A_1/2;r_{1,0}^{(1)})),\underline{hash(E(A_1/2;r_{1,1}^{(1)}))}) & (\underline{hash(E(A_1/2;r_{1,0}^{(2)}))},hash(E(A_1/2;r_{1,1}^{(2)})))\\\underline{(hash(E(A_2/2;r_{2,0}^{(1)})),hash(E(A_2/2;r_{2,1}^{(1)})))} & \underline{(hash(E(A_2/2;r_{2,0}^{(2)})),hash(E(A_2/2;r_{2,1}^{(2)})))}\end{bmatrix}_{2\times 2}$$

$$HBSM'=\begin{bmatrix}(hash(E(A_2;r_2)\boxed{\div}E(A_1/2;r_{1,1}^{(1)})),\underline{hash(E(A_1/2;r_{1,1}^{(1)}))}) & (\underline{hash(E(A_1/2;r_{1,0}^{(2)}))},hash(E(A_2;r_2)\boxed{\div}(E(A_1/2;r_{1,0}^{(2)})))\\\underline{(hash(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,1}^{(1)})),hash(E(A_2/2;r_{2,1}^{(1)})))} & \underline{(hash(E(A_2/2;r_{2,0}^{(2)})),hash(E(A_2;r_2)\boxed{\div}E(A_2/2;r_{2,0}^{(2)})))}\end{bmatrix}_{2\times 2}$$

The challenge string is: 1 0

Look at the elements underlined in the two matrixes:

The second row of *HBSM* is the same as that of *HBSM'*.

In the first row, the second element in the first pair and the first element in the second pair are the same.

If this is the case, the voter is convinced that the DRE uses the randomness given by the voting authority. There is no randomness generated by the DRE. If the assistant verifier finds that the matrix *HBSM'* is correctly computed from the encryption $E(A_2;r_2)$ and the randomness $\begin{bmatrix}r_{1,1}^{(1)} & r_{1,0}^{(2)}\\r_{2,1}^{(1)} & r_{2,0}^{(2)}\end{bmatrix}$, the voter is convinced that the DRE submits a vote on the candidate A_2 as her intent with probability 3/4.