An untraceable, universally verifiable voting scheme

Michael J. Radwin
December 12, 1995

Seminar in Cryptology
Professor Phil Klein

Abstract

Recent electronic voting schemes have shown the ability to protect the privacy of voters and prevent the possibility of a voter from being coerced to reveal his vote. These schemes protect the voter's identity from the vote, but do not do so unconditionally. In this paper we apply a technique called *blinded signatures* to a voter's ballot so that it is impossible for anyone to trace the ballot back to the voter. We achieve the desired properties of *privacy*, *universal verifiability*, *convenience* and *untraceability* at the expense of *receipt-freeness*.

I. Properties of electronic voting

The traditional process of voting in local and national elections is cumbersome because a voter must appear in person at a polling place to cast his vote. Two recent proposals for electronic voting protocols attempt to remove this burden while providing a private and secure mechanism.

In their paper *Receipt-Free Mix-Type Voting Scheme,* Kazue Sako and Joe Kilian devise what they believe to be a "practical solution to the implementation of voting booth" (Sako 393). Rosario Gennaro proposes in *A Receipt-Free Election Scheme Tolerating a Dynamic Coercer* what he believes to be some "practical" assumptions that improve upon the Sako-Kilian scheme and similar protocols (Gennaro 1).

The reader is expected to be familiar with public-key cryptosystems such as RSA or ElGamal and digital signature schemes. She is also expected to have some familiarity with the number theoretic properties of primes and discrete logs.

1 Properties of the Sako-Kilian and Gennaro schemes

1.a Privacy

Current proposals for electronic voting protocols describe several properties of *privacy* and security. First and foremost, a protocol must ensure that votes are private. *Victor the voter* must be sure that any third party cannot determine who he voted for. That is, when Victor submits his vote over a communication channel, he assumes that a malicious *eavesdropper Eve* is listening. In order to achieve privacy, the voting protocol must employ some form of encryption such as a public-key cryptosystem. This privacy

depends on the assumption that it is computationally infeasible for Eve to decrypt Victor's encrypted vote.

1.b Individual and universal verifiability

The Sako-Kilian and Rosario Gennaro proposals describe the property of *individual verifiability*, the ability for Victor to verify if his vote was received properly (Sako 395, Gennaro 7). Victor would desire this property because it proves to him that the voting authority has counted his vote, and gives him some evidence if he needs to levy a complaint because his vote was lost. Individual verifiability allows only Victor to check for the correct receipt of his ballot. Because each voter must check his or her own vote, an auditor would have to contact and receive the cooperation of every voter to audit the election. *Universal verifiability* allows "any voter or interested third party to at a later time verify that the election was properly performed" (Sako 394). Only with universal verifiability can an audit be performed easily, so this property is desired as long as it does not incur too substantial of a cost (Sako 395).

1.c Receipt-freeness

Sako-Kilian and Gennaro credit Josh Cohen Benaloh and David Tunistra with introducing the first *receipt-free* protocol for electronic voting (Sako 393, Gennaro 1). Benaloh and Tunistra showed that other protocols give Victor a *receipt* for his vote, allowing him to later prove to another party that he voted a certain way. Victor could use his receipt to sell his vote, or he could be coerced under some threat into revealing his vote to a third party (Sako 393, Gennaro 1-2).

A voting protocol that does not give Victor such a receipt (and therefore makes selling votes and coercion impossible) is called *receipt-free.* Sako and Kilian achieve this receipt-free quality by using a secure, private communication channel through which the voting authority can send Victor a message (Sako 394). Gennaro achieves the same goal by a different physical assumption: Victor has secure hardware that does "oblivious probabilistic encryption" -- a smart card, which is an electronic encryption device that does not reveal the random numbers it generates (Gennaro 2).

2 Two desired properties: convenience and untraceability

In this section, we introduce two properties of electronic voting that are not addressed by Sako-Kilian or Gennaro. The issues of *convenience* and *untraceability* are desirable if an electronic voting protocol is to replace the traditional mechanism.

2.a Convenience

Sako and Kilian preface their proposal by stating that "the ultimate goal of secure electronic voting is to replace physical voting booths" (Sako 393). Traditional voting places a burden on citizens because they must be at the appropriate polling place in order to vote in a physical voting booth. This inconvenience may affect voter turnout: according to a report from the Population Division of the Bureau of the Census, less than 45 percent of U.S. citizens aged 18 years or older reported voting in the November 1994 election (Census). Electronic voting has the potential to greatly affect voter registration and turnout if the process of voting can be made more *convenient*.

An electronic voting scheme which does not require Victor's presence in a physical voting booth would remove this inconvenience and is therefore desirable. A protocol for voting which allows Victor to vote from any one of several networked polling locations would be superior to the current system but still inconvenient. A greater degree of convenience is achieved when Victor is able to vote from any networked location such as a telephone, ATM machine, or interactive-television set. Ideally, Victor should not require any external device that interacts with the existing networked device. A lesser degree of convenience than that of traditional voting results if such a device is required.

2.b Untraceability

Another desired property is the *untraceability* of a vote. That is, if Victor submits a vote, a second party (the voting authority) or third party (Eve) should be unable to trace the vote back to him. Even after decryption, the voting authority should be unable to determine the origin of a given vote. It should be able to verify that a vote has come from a valid voter, but it should not be able to discover which one; Victor's anonymity would be preserved. Such untraceability is desired because it mimics the behavior of conventional voting protocols.

3 Implementation of these properties

The privacy property exhibited by Sako-Kilian and Gennaro should be a given. It is not hard to realize this requirement; the protocol would simply require that votes be encrypted with the voting authority's public key using a public-key encryption scheme such as RSA or ElGamal. We discuss the decryption process below.

In order to provide receipt-freeness, Sako-Kilian assumes a secure communications channel. To realize this requirement, Victor must vote at a designated polling place that is known to have a secure channel to the voting authority. Such a requirement conflicts directly with the desired convenience property. Thus, the Sako-Kilian mechanism to achieve receipt-freeness is unsatisfactory. The Gennaro receipt-freeness mechanism is also inconvenient because it requires that each voter posses a tamper-proof smart-card. However, such a smart-card could have a modem in it which would allow Victor to vote from any phone. Using such a smart-card, Victor would sacrifice the inconvenience of using such a device in exchange for the convenience of voting from any phone.

The universal verifiability property described by Sako-Kilian and Gennaro is desired because it also mimics traditional voting practices (easy audits and confirmation that a voter has voted). As Sako and Kilian suggest, universal verifiability can be implemented as an extension to the individual verifiability scheme described in Chaum's mixing technique for electronic mail (Sako 394-395, Chaum81 3-6).

Untraceability is computationally possible through a technique called *blinded signatures,* invented by David Chaum (Chaum92, Chaum88). Blinded signatures, used by Chaum in his untraceable electronic cash scheme, allow a party to digitally authenticate a message without knowing the contents of the message (Chaum92 3). We propose a voting protocol based on blinded signatures in section II of this paper.

II. An election protocol utilizing blinded signatures

4 Blinded signatures

4.a Chaum's electronic coin scheme

Blind signatures were proposed by Chaum in *Untraceable Electronic Cash* as a technique realize untraceable electronic coins. The scheme relies on the bank creating a number system where only it can compute cube roots. A coin that Alice would want to spend starts off as a number x that acts like a serial number for a bill. The number x is a 100-digit number that Alice chooses at random, so there is very low probability someone else will pick the same serial number (Chaum92 2).

This serial number needs to be digitally signed by the bank so that the bank will later recognize it as currency that someone was authorized to spend. However, in order to protect her anonymity, Alice will multiply x by the cube of a another random number, r^3. This extra random number is called the *blinding factor* because it "hides" the value of x from the bank. This blinding factor, according to Chaum, is unconditionally untraceable to Alice: "Even if the bank had infinite computing power, they couldn't find out because it contains just as much r information as $[x]$ information" (Chaum94 2).

Each coin is a pair $(x, f(x)^{1/3} \pmod n)$ where f is a one-way function and n is some composite whose factorization is known only to the bank (Chaum88 319). Since only the bank knows the factorization of n, only it can compute cube roots modulo n, so the cube root acts as a digital signature from the bank (Chaum94 2) The basic coin issuing and spending protocol described in *Untraceable Electronic Cash* is:

1. Alice chooses a random x and r, and supplies the bank with $B = r^3 f(x) \pmod n$.
2. The bank returns the third root of B modulo n: $r * f(x)^{1/3} \pmod n$ and withdraws one dollar from her account.
3. Alice extracts $C = f(x)^{1/3} \pmod n$ from B [by dividing by the blinding factor r].
4. To pay Bob one dollar, Alice gives him the pair $(x, f(x)^{1/3} \pmod n)$.
5. Bob immediately calls the bank, verifying that this electronic coin has not already been deposited. (Chaum88 319-320)

4.b Preventing double-spending

Because the coins in Chaum's scheme are just numbers, Alice could easily spend a coin twice by making a copy of it and spending it at another vendor. In order to detect an occurrence of double-spending, Chaum introduces another stage in the algorithm which requires Alice to answer a random set of questions about her coin. The responses to the random set of questions, sent to the bank, reveal some information about the coin but do not compromise her unconditional privacy (Chaum92 3).

However, if Alice attempts to spend a copy of her coin at another vendor, there is a very high probability that the information she reveals to the second vendor will combine with the information she revealed to the first vendor to show that it was Alice who attempted

to spend a coin twice. That is, there is a very high probability that Alice will be caught as a double-spender (Chaum88 322). We will apply this technique below to recognize double-voters. The reader is referred to *Untraceable Electronic Cash* if she cares to read the specifics of the Chaum scheme.

4.c Anonymous voting and double-vote recognition

In order to participate in an election, Victor is must take part in two phases. In the first phase of voting, Victor is required to *register* with the *voting authority* so that he can obtain a digitally signed *numerical pseudonym*. The pseudonym will be constructed from his Social Security Number to trace his vote back to him in case he double-votes, but will be not be recognizably associated with his identity if he only votes once. After he receives his blinded digital pseudonym, he removes the blinding factor and uses the pseudonym in the second phase when he submits his ballot to the voting authority.

4.d Registration phase

In the registration phase, Victor goes to the office of the voting authority (such as the county Registrar of Voters) to obtain a digital pseudonym. As above, the authority publishes the RSA modulus n but does not reveal its factorization. The authority also determines a security constant k that will be used in the double-vote prevention algorithm (Chaum88 320).

After identifying himself as an eligible voter to the authority, Victor and the authority interactively determine a numerical pseudonym. Let Victor's Social Security Number be u and define XOR to be a bitwise exclusive-or. "Let f and g be two argument collision-free functions; that is, for any particular such function, it is infeasible to find two inputs that map to the same point. We require that f be 'similar to a random oracle'. For unconditional traceability we also require g to have the property that fixing the first argument gives a one-to-one (or c to 1) map from the second argument *onto* the range" (Chaum 320)

To get a pseudonym, Victor performs the following exchange with the authority (modified from Chaum 321):

1. Instead of producing a single x and r, Victor's smart card produces the random numbers a_i, c_i, d_i, and r_i, $1 <= i <= k$, randomly (mod n).

2. Victor sends the voting authority k *blinded candidate* numerical pseudonyms which we call B.
Let $B_i = r_i^3 * f(x_i, y_i) \bmod n$ for $1 <= i <= k$
where
$x_i = g(a_i, c_i)$ $\quad$ $y_i = g((a_i \text{ XOR } u), d_i)$

3. The authority picks a random subset of $k/2$ of the indices candidates and asks Victor to show how he arrived for his values of f and g. For convenience of notation, we will assume that the authority asks for indices $R=\{k/2 + 1, k/2 + 2, \ldots, k\}$.

4. Alice displays a_i, c_i, d_i, and r_i,for each i in R. The authority verifies that Victor computed the respective values for x_i and y_i correctly.

5. After verifying the validity, the authority sends to Victor:

$$\prod_{i \notin R} Bi^{1/3} = \prod_{1 \le i \le \frac{k}{2}} Bi^{1/3} mod n$$

6. Victor can then easily extract the numerical pseudonym

$$P = \prod_{1 \le i \le \frac{k}{2}} f(xi, yi)^{1/3} mod n$$

4.e Vote submission and double-vote recognition

Now that Victor has obtained a valid pseudonym *P*, he can submit his vote. Let *w* represent his vote and let *w* be 0 for a "no" vote and let *w* be 1 for a "yes" vote. When Victor wants to submit his vote, he and the voting authority do the following (modified from Chaum 321-322):

1. Victor prepares a ballot as the pair (*w*, *P*) and encrypts it with the voting authority's public key. He sends the encrypted ballot to the voting authority.

2. The voting authority decrypts the message. It then chooses a random binary vector *Z* with elements $z_1, z_2, ..., z_{k/2}$ and sends it to Victor. This is a challenge to Victor to prove that *P* is valid.

3. Victor responds as follows, for all 1 <= *i* <= *k/2*:
   a. If $z_i = 1$, Victor reveals to the voting authority a_i, c_i, and y_i
   b. If $z_i = 0$, Victor reveals to the voting authority x_i, $(a_i$ XOR *u)*, and d_i

4. The voting authority can verify that *P* is of the proper form and that Victor's responses correctly fit the *P* he sent in.

5. It also checks to see if it has received a ballot from a voter with pseudonym *P* before. If it has seen *P* before, it can with high probability determine which voter attempted to double-vote (see below).

6. If *P* is formed properly by (4) and it has not seen *P* before in the ballots it has received, the voting authority declares *w* to be a valid vote and adds it to the tally. The voting authority stores *Z*, *P*, and Victor's responses to the *Z* challenge in case.

In part (3) of the above protocol, Victor reveals at random exactly one-half of each pair that is required to compute *f*. Victor's anonymity is preserved because a value of *u* cannot be derived by the voting authority until it has both a_i and $(a_i$ XOR *u*). However, if Victor attempts to vote twice, there is a high probability that z_i will be 0 for one vote and 1 for another, for some *i*. That is, with high probability, Victor will send a_i to validate one of his votes and $(a_i$ XOR *u)* to validate another. The voting authority records can verify this because it recorded *Z*, *P*, and Victor's responses to the *Z* challenge in (6) above.

If Victor attempts to double vote, the authority will witness the use of *P* twice. In this case, it compares the two *Z* vectors until it finds a complimentary pair in a given index *i*. It then looks at the responses to the *Z* challenge, and computes a_i XOR (a_i XOR *u)*. Due to the properties of XOR the expression yields *u*, Victor's real identity. When the authority

has determined that Victor has double-voted, it can disqualify his ballot and possibly take legal action. The probability that such a complimentary pair of indices in Z exists is exponential in the size of k. The voting authority needs only to make k sufficiently large to catch Victor double-voting with high probability.

5 Individual and universal verifiability

5.a Individual verifiability

Sako and Kilian achieve individual verifiability by using the scheme described in Chaum's mixing technique for electronic mail (Chaum81). Chaum starts with the idea that mail messages should be sent through a trusted computer called a "mix" which strips identifying information from its inputs and re-sends the messages to their destinations. If Bob wanted to send a message M to Alice at email address A, he would first encrypt M with Alice's public key E_A. Then, Bob sends both Alice's address and the encrypted message to Mix 1. In order to prevent Eve from observing the fact that he is sending a message to Alice, Bob encrypts the pair $(A, E_A(M))$ with Mix 1's public key. He sends the message $E_1(A, E_A(M))$ to Mix 1 (Chaum81 3-4).

Chaum uses several techniques to hide the origin and path of messages. First, he proposes adding some random bits R_1 to the message sent to Mixer 1 so that the message is less easily guessed. When Mix 1 receives the message $E_1(R_1, A, E_A(M))$, it just ignores the term R_1 after decryption (Chaum81 4). He proposes that mixes send out their messages in permuted batches. A mixer outputs messages of similar size in lexicographically ordered a batches to remove a possible correspondence between arrival and departure time from the mixer (Chaum81 4). Using a "cascade", or a series of mixes, to ensure that messages are further shuffled. When the message finally reaches its message, every mix in the cascade can send back a proof to Bob that the original message reached its Alice successfully (Chaum81 5).

5.b Universal verifiability

The Sako-Kilian protocol uses a discrete-log public-key cryptosystem for the mixing (Sako 395). Gennaro applies a key escrow technique known as *secret-sharing* to a modified version of the ElGamal encryption scheme (Gennaro 7). The spirit of both universal verifiability proofs is the same: In Sako-Kilian, each mix is required to prove that it processed all of its ballots properly (Sako 396-398). In Gennaro, each of the four phases of the protocol comes with a publicly verifiable proof of correctness (Gennaro 9-10). Both protocols require that the final collection agency publish a verifiable tally (Gennaro 10, Sako 397-398).

In the anonymous scheme we present here, such mixing is not required for universal verifiability because there is no mathematical correlation between a voter's true identity and his pseudonym. The voting authority could simply publish a tally of "yes" and "no" ballots along with each corresponding pseudonym P. In order to achieve individual verifiability, Victor could find his own P in the list and confirm that his ballot was tallied properly. Universal verifiability would follow the same mechanism because the list is public voting authority can prove that it correctly received every P on the list of

received ballots.

Unfortunately, there is a problem with such a universal verifiability scheme. Victor now has a receipt for his vote; he can show the random values a_i, c_i, d_i, and r_i he used to generate B and then show the correlation to the resulting P he received from the voting authority. The property of receipt-freeness is no longer preserved.

III. Conclusions

In an attempt to apply the properties of *convenience* and *untraceability* to the Sako-Kilian and Gennaro voting protocols, we sacrificed the desired property of *receipt-freeness* while maintaining the properties of *privacy* and *universal verifiability.* A voter Victor who uses our protocol would maintain complete security and anonymity, but could be coerced into revealing his vote because he maintains a receipt.

References

Chaum, David. "Achieving Electronic Privacy," *Scientific American,* pp. 96-101, August 1992. <http://www.digicash.com/publish/sciam.html>

Chaum, David, and Naor, Moni, and Fiat, Amos. "Untraceable Electronic Cash," *Advances in Cryptology: CRYPTO '88,* Lecture Notes in Computer Science, vol. 403, pp. 319-327, Springer-Verlag, New York, 1988.

Chaum, David, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM,* vol. 24 no. 2, February, 1981. <http://draco.centerline.com:8080/~franl/crypto/chaum-acm-1981.html>

Chaum, David, "Digital Money", lecture, *Doors of Perception 2, @HOME Conference,* Amsterdam, November 1994. <http://mmwww.xs4all.nl/Doors/Doors2/Chaum/Chaum-Doors2-E.html>

Bureau of the Census. "Characteristics of the Voting-Age Population Reported Having Registered or Voted: November 1994," *Voting and Registration Supplement to the November 1994 Current Population Survey,* p. 1, June 1995, <http://www.census.gov/ftp/pub/population/socdemo/voting/profile/ptable1.txt>

Gennaro, Rosario. *A Receipt-Free Election Scheme Tolerating a Dynamic Coercer (with Applications to Key Escrow),* Massachusetts Institute of Technology, November 1995.

Sako, Kazue and Kilian, Joe. "Receipt-Free Mix-Type Voting Scheme: A Practical Solution to the Implementation of a Voting Booth," *EUROCRYPT '95,* vol 921, *Lecture Notes in Computer Science,* pp. 393-403, Springer-Verlag, 1995.