Secret-Ballot Receipts: True Voter-Verifiable Elections

A new kind of receipt sets a far higher standard of security by letting voters verify the election outcome—even if all election computers and records were compromised. The system preserves ballot secrecy, while improving access, robustness, and adjucation, all at lower cost.

David Chaum

Current electronic voting machines at polling places don't give receipts. Rather, they require prospective voters to trust them—without proof or confirming evidence—to correctly record each vote and include it in the final tally. Receipts could assure voters that their intended votes are counted. However, receipts have so far not been allowed because of the "secret ballot" principle, which forbids voters from taking anything out of the polling place that could be used to show others how they voted. The reason for this is to prevent schemes that could improperly influence voters, such as vote selling and various forms of coercion.

Introduced here is a fundamentally new kind of receipt. In the voting booth, the voter can see his or her choices clearly printed on the receipt. After taking it out of the booth, the voter can use it to ensure that the votes it contains are included correctly in the final tally. But, because the choices are safely encrypted before it is removed from the booth, the receipt cannot be used to show others how the voter voted.

The receipt system can be proven mathematically to ensure election integrity against whatever misbehaving machines or people might do to surreptitiously change votes. This level of integrity should enhance voter satisfaction and confidence and positively impact participation.

The system also eliminates the need for trusted voting machines, which typically use proprietary "black box" technologies. It can run with published code on standard PCs, allowing significantly lower cost and higher quality. The receipts also improve robustness, currently achieved by costly proprietary hardware redundancy in storing and transporting votes, not only because failures can be detected at the polls in time to prevent lost votes, but also because the votes that receipts contain can be counted no matter what happens to the machines. Moreover, open-platform hardware, instead of being stored in special warehouses most of the time, could even be used for various purposes year-round, for example in schools and libraries.

The inability of the current approach to reconcile secrecy and security needs has also led to functionality problems. The new US Federal requirement for provisional ballots—ballots cast by individuals whose names don't appear on the registration list—means separate handling and counting, singling provisional ballots out for reduced privacy protection. Just as the system presented here can seamlessly include all such votes, it can lift the requirement that voters vote from their home precinct, ensuring access while improving convenience and turnout. (It even makes interjurisdiction voting workable.) Courts can also surgically add or remove the votes of particular fine-grained categories of voters; their inability to do so today forces them to call revotes, throw out all ballots, or determine winners themselves.

Voting with the new approach

After you input your choices using a touch screen or other input means, with the new approach, a small device that looks like a cash register printer generates a printout (part of which will become your receipt). The printout lists the names of the candidates you chose along with their party affiliations and offices sought, as Figure 1 shows, as well as your vote on any ballot questions. Included are allowed write-ins and other choices, such as with straight-party voting and prioritized and

Figure 1. An example part of a ballot printout listing a candidate selected. In addition to being able to include the candidate's name, party affiliation, and office sought, the printout can also include other types of contests and various graphics options.

weighted votes. The printout might also include graphics, such as a voter's handwritten choice of candidate, party symbols, or (someday) photographs such as some countries use. It might also alert you to contests or questions you skipped and serves as the single summary of your vote. After printing your votes, the machine prompts you to review the printout still in the printer and accept it, giving you the opportunity to amend your vote and generate a new printout.

Generating a receipt

If you agree with the printout, the machine asks you to indicate whether you wish to keep the top or the bottom layer of it. The printer differs from ordinary receipt printers because it simultaneously prints separate but aligned graphics on both the top and bottom sides of the strip. After you've indicated your choice of layer, the machine prints the final inch of the form. (The voter choosing which layer only after the main part is printed is key to keeping the system honest.) It then automatically cuts off both layers, still laminated together, and releases them to you. Figure 2 shows the laminated last inch of the printout.

As you separate the layers, the image of the votes becomes an unreadable and seemingly random pattern of tiny squares printed on each of two layers of translucent plastic material. Neither layer is readable on its own—the light passing through the sandwiched layers only where neither layer has printing is what makes your choices visible. Still, each layer separately and safely encodes your vote exactly as you saw it.

The last inch of the printout is different because its layers have messages that are readable after the layers are separated, as Figure 3 shows. The layer you select to keep as your receipt bears a message such as, "Voter keeps this privacy-protected receipt layer" (Figure 3a), whereas the other layer might state, "Voter must surrender this layer to poll worker" (Figure 3b).

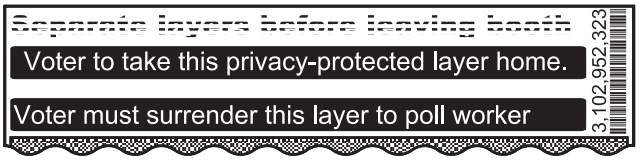
Verifying your vote

As you leave the polling place, you give the poll worker the layer marked for surrender. For your protection and as you watch, the poll worker checks that it's the correct layer and destroys it in a small, transparently housed paper shredder. You keep the other layer as your receipt. The voting machine keeps an electronic version of this same final receipt until it successfully sends it in for posting on the official election Web site. The bits on the shredded paper layer are also "shredded" electronically—that is, the only things that remain of your vote are your physical layer and, in the machine, a digital version of that same image.

(One way to handle voters that refuse to surrender layers is for the exit shredder—based on its reading of the

Figure 2. Last inch of the printout before the two laminated layers are separated.

(a)



(b)

Figure 3. Last inch of the printout after it's separated: (a) the receipt (the layer the voter selects to keep) and (b) the layer that's shredded before the voter leaves the polling place.

ballot serial number barcode, which additionally prevents shredding the wrong layer and allows spoiling of "missing" receipts—to give a sticker with a key needed to decrypt the last inch. This also lets poll workers issue the other last inch to voters claiming their choice of layer was switched.)

You can safely show your receipt to anyone, including political, governmental, public interest, or media organizations. Outside the polling place, for example, a group such as the League of Women Voters might offer to check your receipt. They simply scan it with a handheld scanner and let you know immediately that it's authentic and correct (by subjecting the receipt's printed image and its coded data to a consistency check and later ensuring that it's correctly posted online when it should be, all of which is detailed later). An invalid receipt would irrefutably indicate incorrect operation of election equipment, although a second scanner could readily dispel a false alarm.

When the polls close, the polling place sends only the digital form of the receipts (not the shredded layers or cleartext votes), electronically or by transport of, say, a CD.

Election Web site

If you wish, you can find the page on the official election Web site that includes your receipt by entering the receipt's serial number. You could then check that your vote was posted correctly—for example, by printing the posted receipt and overlaying it with your original receipt and checking that they are identical. (You need not run consistency-checking software, because anyone can do this for all posted receipts, as discussed later.) You could also provide the original or its image by fax or photocopy to others for checking.

At some point after the polls close, the definitive set of receipts to be counted—the *receipt batch*—is posted on the Web site along with attesting signatures. The election's final output—the *tally batch*—is similarly posted. It contains the same number of items as the receipt batch, but each is a readable plaintext image of the ballot exactly as the voter saw it in the booth. (Using simple software, anyone can compute the totals from the tally images.) To protect privacy and ballot secrecy, the tally batches are in a random order, thereby hiding the correspondence between receipts and ballot images.

To ensure that a one-to-one correspondence does in fact exist between the batches—that is, that no ballots were inserted, deleted, or changed—the system uses a kind of audit of a chain of intermediate batches between the receipt batch and tally batch. After creating and publishing the intermediate batches, the system decrypts randomly chosen samples from them. These samples are chosen so as not to reveal enough to compromise privacy. They reveal enough, however, that checking them against the published batches effectively thereby checks that the correct one-to-one correspondence holds. Anyone can do this checking by running a simple, open-source program that they can download from any of multiple suppliers or even write themselves. The program can also check the consistency of each receipt batch entry. Such a suite of checks can convince anyone that the receipt batch correctly yielded the tally batch.

Receipt system

The system introduced in this section is detailed in the "More formally" sidebar (on page 44), which in turn serves as a basis for the "Proof sketches" sidebar (on page 46).

Properties

The receipt system ensures several properties.

First, if your receipt is correctly posted, you can be sure (with acceptable probability) that your vote will be included correctly in the tally. A receipt that isn't properly posted is physical evidence of a failure of the election system, and a refusal by officials to post it is an irrefutable admission of a breakdown in the election process.

In addition, no one can decode your receipt or otherwise link it to your vote except by breaking the code or decrypting it using all the secret keys, each of which is assigned to a different trustee.

Even if all the election computers were compromised and running colluding malicious software (even having access to unlimited computing power), there are only three ways that a system could change a voter's correctly posted ballot without direct detection:

- It could print an incorrect layer, gambling that the voter will choose the other layer.
- It could use the same serial number for two different receipts, hoping the two voters choose the same layer.
- It could perform a tally process step incorrectly, taking the chance that the step will escape selection during audit.

For each ballot and with any of the three approaches, the chance that it would go undetected is one half. Thus, the chance that two ballots could be changed without detection of at least one is only a quarter, three ballots without a single detection an eighth, and so on. Changes in just 10 ballots will avoid any detection fewer than one in 1,000 times, and changes in 20 ballots will avoid detection fewer than one in 1,000,000 times.

In practice, many voters will not check that their receipts are posted or even have others check them. For example, in a large election, if just 10 receipts are changed and only 5 percent of receipts are checked at random, the chance of detections is 50 percent. But in close elections in which a small number of ballots matter, a sufficiently high percentage of ballots would presumably be checked at least after the results were pub-

lished. For example, if 100 votes would have changed the outcome in a large election, 5 percent of receipts checked would be enough to catch cheating all but one in 1,000 times.

Receipt encoding

What makes the laminated layers readable and the separated layers meaningless is the mutual relationship of the patterns printed in black on each translucent plastic layer. The printing on both layers is divided into a grid of squares, or *pixel locations*. Each pixel location is printed with one of two *pixel symbols*, like a large, filled-in tic-tac-toe board. The two pixel symbols are reverses of each other: where one is clear, the other is black, and vice versa. When two different pixel symbols are aligned one directly on top of the other as they are when laminated, any clear spot on one is blocked by black on the other, making the lamination appear totally opaque. When the same pixel symbol is printed on both layers and the symbols are aligned, all the clear parts are directly over each other and light can thus pass through the laminate. Figure 4 shows layers with both the same and different symbols overlaid.

This technique can be used to encode information on one sheet so only someone with a second sheet can read it, the application that Moni Naor and Adi Shamir first proposed it for.[1] It's useful to associate names with the two sheets: I'll call the first "white" and the second "red" (but these colors have no more graphic significance than that you might tint the two translucent sheets to distinguish them). Each sheet is divided into a grid of pixel locations, and each pixel location has a pixel symbol printed on it. When the two sheets are laminated together, the grids line up exactly: each pixel location on one sheet has a *paired* pixel location at the same coordinates on the other sheet so the two are exactly one on top of the other. First you choose the pixel symbols for the white sheet totally at random. Now to encode your message in the laminate, you simply choose each of the symbols of the red sheet accordingly: If you want light to shine through for a pixel location when laminated, you choose the same pixel symbol as its paired pixel on the white sheet; if you don't want light to go through at that location, you choose the other symbol.

Most current printing technologies print ordinary text by creating a grid of pixel locations in which some are printed fully with black ink while others get no ink. For the present system, instead of leaving the background without ink, the system pairs nonmatching (that is, opaque) pixel symbol combinations; instead of using full black ink for letters, the system pairs matching (that is, partly clear) pixel-symbol combinations, creating gray letters that depend on backlighting for brightness. Figure 5 illustrates the differences between the techniques.

(The system can use modified direct thermal printers, like those deployed at most checkout counters. These printers have two to three times the resolution needed here, but this can be used to frame pixels and forgive mechanical alignment errors between the ceramic print heads that would run the width of the paper on top and bottom. A clear "fugitive" adhesive laminates the layers and isn't sticky when the layers are delaminated.)

When the receipt layers are still laminated, the voter's choices are thus printed in a gray made up of half black and half white spots on a black background. This *ballot image* is the visible plaintext summary of the vote accepted by the voter.

Because the vote should be encoded in each layer separately, both layers need some red pixels. Swapping two paired pixel symbols between the layers leaves the laminate visually unchanged. So pairs in half of the pixel locations, say, in a checkerboard pattern, are swapped. If the pixels were tinted, instead of separate red and white layers,

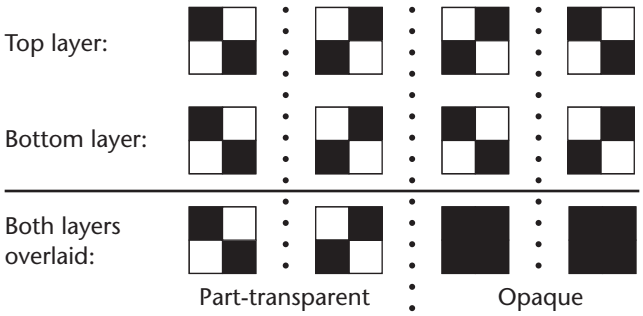


Figure 4. The two pixel symbols, separate and overlaid. When two different pixel symbols are overlaid, the result is opaque; matching pixel symbols let light through.

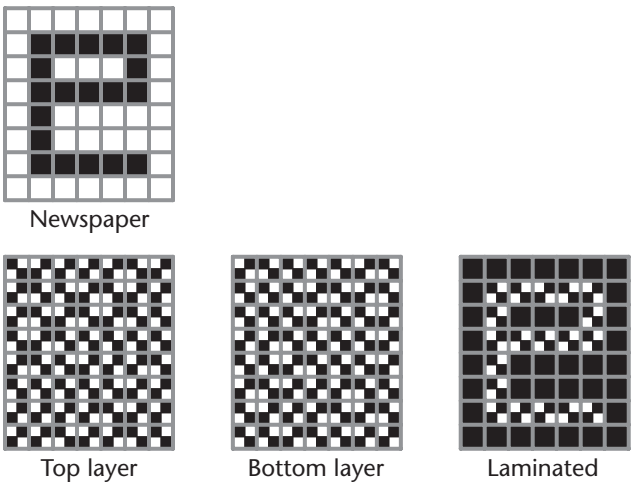


Figure 5. The letter "e" in (a) standard printing and (b) receipt printing. The receipt printer pairs matching and nonmatching pixel symbols to produce letters and blank space, respectively.

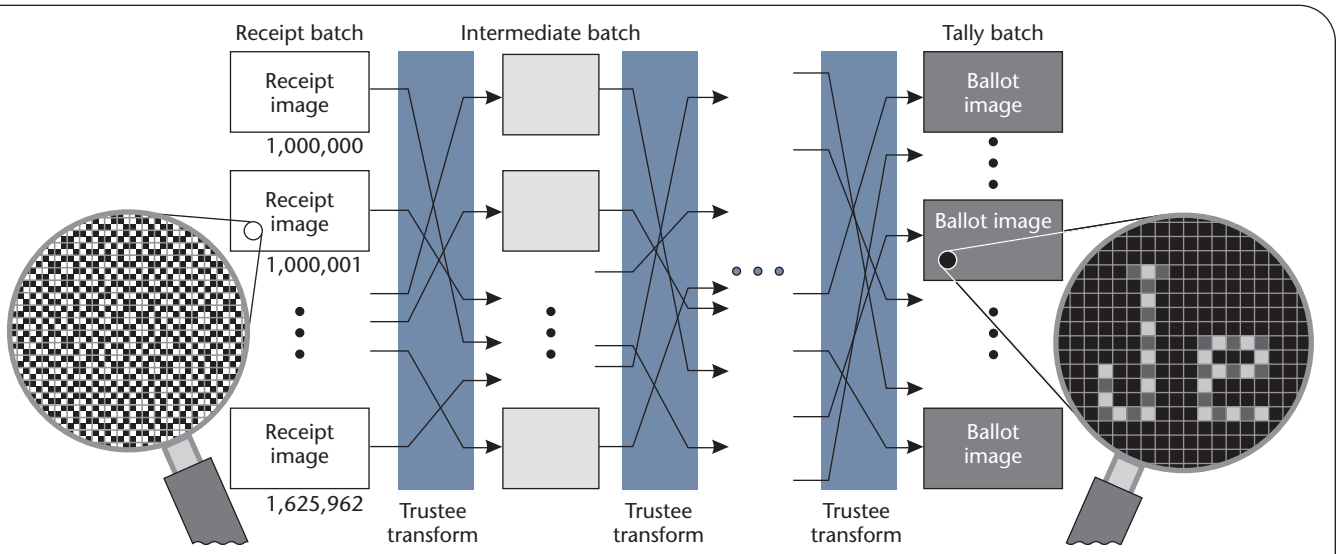


Figure 6. The overall tabulating process. Receipts pass through trustee-operated mixes, which transform them step-by-step into cleartext ballot images to be posted and tallied. Serial numbers (and all but the red half of the pixel symbols) are stripped off in forming the first intermediary batch. Mixes transform by removing a layer of encryption from each input and reordering the inputs in their output. Vertical ellipses indicate batch items not shown; horizontal ellipses indicate additional trustees. (Darker ballot image pixels are inferred from the lighter ones using redundancy in the font.)

each layer would look like the red and white tablecloths in a typical bistro.

The system in effect uses the *one-time pad* coding technique to encrypt the ballot image. Claude Shannon proved this technique to be unbreakable, assuming the key is random.[2] The keys used—the white pixels—aren't random but are believed to be indistinguishable in practice from random except to the set of trustees, who collectively guard ballot secrecy. Thus, if you have only your receipt layer and are staring at a particular white pixel on it, you learn nothing. Similarly, a red pixel only tells you that the lamination would have been partly clear if the paired white pixel matched the red pixel and opaque if it didn't. But knowing nothing about which white pixel symbol was paired means you can't infer anything more about whether the combination was partly clear or opaque.

Receipts should encode the votes exactly as the voter sees them. It's technically possible, however, that the still laminated printout shows one set of choices, but the receipt layer the voter takes encodes other choices. This could occur only if just one layer was invalid. If both layers are invalid, whichever layer the voter takes will fail checking and provide direct evidence of cheating by the system. If only one layer is invalid, and the voter doesn't select it, it won't be checked, just shredded. However, essential to security, as mentioned earlier, is that the voter chooses which layer to take only after the printer finishes printing the votes. Thus, a single invalid layer has essentially a 50-50 chance of being selected by the voter and caught.

Tabulating process

When the polls close, election officials or the courts should resolve any provisionally or otherwise contested voting and then post the receipts to be included in the tabulating process electronically as the official definitive receipt batch. (A preliminary tally formed before contested and provisional ballots are included can, to obscure the provisional/contested votes being adjudicated, omit a random selection of ballots that will be included in the final tally.)

The tabulating process starts with a receipt batch and produces a final tally batch of ballot images. The first trustee produces the first intermediate batch from the receipt batch. The next trustee forms the second intermediate batch from the first intermediate batch, and so forth, until the last trustee forms the tally batch from the last intermediate batch. Figure 6 diagrams this process.

Trustees change the coding and the order of items from each batch in the chain to the next, thus ensuring privacy. Requiring each trustee to release some random samples, establishing that items have been correctly transferred from batch to batch, ensures integrity.

Russian nesting dolls

A Russian nesting doll analogy can illustrate the processing of the input batch of receipts into the tally batch of ballot images. Each batch corresponds to a collection of dolls, each doll to an item in the batch. The receipt batch, for instance, is a collection of outermost "big" dolls, each with all its smaller dolls neatly nested within. The next batch, the first intermediary batch, is similar to the receipt batch but

without the big dolls. This continues to the tally batch: the tiny solid wood innermost dolls. All batches have the same number of dolls, and within a batch the outermost dolls are all the same size and contain their own smaller dolls.

The nesting dolls are like secret agents, each doll holding a unique random code sheet in its hands. The sheet is a grid of pixels printed using the two pixel symbols. Each doll is also physically locked with a combination lock that prevents access to the dolls within. A different secret combination, known only to a single corresponding trustee, unlocks all dolls of a particular size.

Consider the trustee with the secret combination for, say, the 10-inch dolls. To process an individual doll in the batch of 10-inch dolls, the trustee first unlocks the doll using the secret combination and removes its contents, a 9-inch doll. The trustee now has two code sheets, one from the 10-inch and one from the 9-inch doll. The trustee combines the two sheets to produce a new code sheet as follows: for every pixel location where light passes through the two sheets when stacked, one pixel symbol is printed on the new sheet; everywhere no light passes through, the other symbol is printed. (When each of the two pixel symbols is considered a binary digit, 1 or 0, combining any number of sheets is simply adding the values modulo two.) The trustee places the combined code sheet in the hands of the 9-inch doll and destroys the empty 10-inch doll along with both old code sheets.

After likewise processing all the 10-inch dolls into 9-inch dolls with new code sheets, the trustee randomizes their order and outputs them as a batch. The trustee with the secret combination for the 9-inch dolls takes this batch as input, processes it into a batch of 8-inch dolls, and so on.

Coded sheets

A simple way to apply this process to an election starts by forming the sheet held by each big doll differently from all the sheets of the dolls nested within it. Suppose the original doll maker faithfully chooses sheets for all the dolls inside a big doll at random, but makes copies of all the sheets. Instead of keeping these copies on separate sheets, the doll maker combines them into a single sheet for the big doll, one pair of sheets at a time (or all at once using modulo-two addition). This is the "white" sheet for that big doll. Intuitively, it's formed by an initial "adding in" of all the inner sheets' coding, which will be "subtracted out" in stages as the dolls are processed.

Now suppose a voter has one of these big dolls and wants to use it to vote with privacy. The voter determines a red sheet that produces the desired ballot image when optically combined with the doll's white sheet (as previously explained). The voter then shreds the white sheet and gives the doll the red sheet to hold, placing the doll in the initial batch of big dolls. After processing by all the trustees, the final output batch contains the tiny solid wood dolls in random order, each holding a sheet that reveals a ballot image (which is easily seen by laminating with a sheet containing the same pixel symbol copied everywhere). All of the code sheets combined in the white sheet that influenced the red sheet have now been subtracted out.

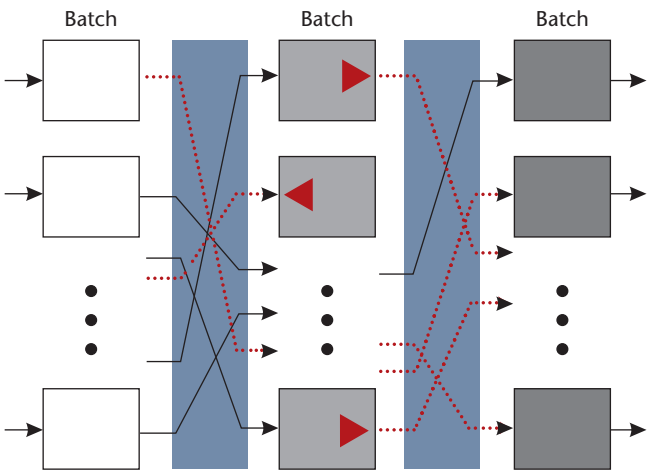


Figure 7. Batch processing by a single trustee. Triangles show the result of the random public draw and the broken lines show links whose details are accordingly released in audit.

To provide integrity, the system must be able to catch any trustee attempting to improperly change dolls or their sheets during processing. The solution will entail requiring trustees to release complete and detailed audit trails of the processing (as videotapes, for example), but only for select dolls.

To allow trustees to release half of the complete set of tapes without compromising ballot secrecy, they each take on the role of processing more than one of the batches, say, two successive batches of the chain. This prevents tracing any tiny doll back to a big doll, even by a collusion of all but one trustee. After processing, a public lottery draw selects half the dolls in the trustees' first input batch, and the trustee releases their videos. Videos of these dolls' second processing wouldn't be revealed (because they might allow linking), but the second-batch videos of the other dolls are revealed. Figure 7 shows an example processing of two such batches by a trustee.

Exact tracing is thus prevented because trustees release only one video per doll for the two adjacent batches. Still, each time a trustee improperly forms a batch item there is a 50-percent chance of it being selected for release, so the odds of being caught stack up just as fast as with cheating by introducing a bad printed layer.

Encryption

Returning to the receipt system, the analogy's red and

More formally

A complete system can be described somewhat more abstractly and formally, much as a typical cryptographic protocol: in terms first of what messages should be exchanged in what order, and then how the parties are to check what they receive. The receipt system has two separate phases: a voting phase and a tally phase.

Voting phase

The voting phase comprises a number of instances, each of which has up to six successive steps:

1. The prospective voter supplies a ballot image B.
2. The system responds by providing two 4-tuples: $<L^z, q, D^t, D^b>$, ("L" is for layer, "q" is for serial number, "D" is for doll, and "z" is for either "t" for top layer or "b" for bottom layer.) Each 4-tuple is printed on a separate transparent layer.
3. The voter verifies (using the printing's optical properties) that $L^t \oplus L^b = B$ and that the last three components of the 4-tuple are identical on both layers.
4. The voter either aborts, and is assumed to do so if the optical verification fails, or selects the top layer $x = t$ or the bottom layer $x = b$.
5. The system makes two digital signatures and provides them as 2-tuple $< s^x(q), o^x(L^x, q, D^t, D^b, s^x(q)>$ ("s" is for seed and "o" is for overall).
6. The voter (or a designate) performs a consistency check to ensure that the digital signatures of the 2-tuple check, using agreed public inverses of the system's private signature functions s^x and o^x, with the unsigned version of the corresponding values of the selected 4-tuple (as printed) on the selected layer, and that $s^x(q)$ correctly determines D^x and the half of the elements of L^x that it should determine.

More particularly, let the relationships between the elements of the 4-tuples and the 2-tuple be as follows: The red bits R^z and white bits W^z (both m by $n/2$ where n is even) determine the m by n binary matrices L^z in a way that depends on whether $z = t$ or $z = b$: $L^t_{i,2j-(i \bmod 2)} = R^t_{i,j}$, $L^t_{i,2j-(i+1 \bmod 2)} = W^t_{i,j}$, $L^b_{i,2j-(i+1 \bmod 2)} = R^b_{i,j}$, $L^b_{i,2j-(i \bmod 2)} = W^b_{i,j}$, where $1 \le i \le m$ and $1 \le j \le n/2$. The ballot image and the paired white bits of the opposite layer y determine the red bits: $R^x \oplus W^y = B^x$.

The cryptographic pseudo-random sequence functions h and h' (whose composition yields binary sequences of length $mn/2$) determine the white bits from the signature on the serial number as follows: $W^z_{i,j} = (d^z_k \oplus d^z_{k-1} \oplus \ldots \oplus d^z_1)_{(mj-m)+i}$, where $d^z_l' = h(s^z(q), l)$ and $d^z_l = h'(d'^z_l)$. The d'^z_l also forms the "dolls" using the public-key encryption function e_l, whose inverse is known to one of the trustees: $D^z_l = e_l(d'^z_l \ldots e_2(d'^z_2, (e_1(d'^z_1))$, where $1 \le l \le k$ and, for convenience, $D^z = D^z_k$. (Separate h and h' are for improved efficiency with large ballots.)

white sheets correspond, of course, to a ballot's red and white pixels (although without checkerboarding). The analog of a lockable wooden doll is *public-key encryption*, in which anyone can encrypt a message using a published public key, but only the holder of the corresponding private key, the trustee, can decrypt it. Thus any voting machine can in effect be a doll maker and successively form the layers of a digital doll using published keys, but only trustees can strip off the respective layers. (Various known redundancy and key-sharing techniques provide resiliency in case some trustees don't participate.) With encryption as the mechanism instead of a videotape, in effect only the code sheet originally held by the output doll must be released. (It's easy to check that applying the public key to the combination of this original sheet and the output doll results in the input doll.)

The initial printout in the voting booth actually uses two dolls. One of these is checked completely by being reconstructed from values printed on the last inch of the receipt layer and then not used further. The other doll and its checkerboard half of the red pixels create a "duo" that travels together through the chain of batches in the tally process. Such duos make up all batches. Trustees process each batch by removing a layer of encryption from the duo's doll and applying the revealed digital sheet to the duos pixels. By the time the duo reaches the tally batch, nothing is left of the dolls, and the pixels have become a readable plaintext ballot image.

(Dolls that include error correction are printed on the layers of plastic in a special way. A copy of both such dolls is printed on one layer; on the opposite layer the same image appears but with the pixel symbols reversed. This creates a uniform opaque background around the votes whose absence would be easily noticeable to voters, ensuring that each layer has identical copies of both dolls.)

What codes to use?

Digital signatures are printed in the barcode on the last inch of the receipt layer. Such signatures have legal standing in many countries, and are considered irrefutable proof of the signed message's origin. A verifier outside the polling place can scan your receipt to immediately check, among other things, that its signature is valid, that an authorized voting station generated it, and that it correctly covers all the data printed. If the signature doesn't pass, the physical receipt is direct evidence of system failure. If the receipt does check, however, it cannot be credibly denied a place in the definitive receipt batch.

Cryptographic techniques are classified as either *unconditionally secure* or *computationally secure*. The former, like the one-time pad with random key, cannot be broken, even if an adversary were to apply infinite comput-

Tally phase

The tally phase takes its input batch from the outputs of an agreed-on subset of voting instances reaching step 6. For each such instance, only half of L^X and all of D^Y are included in the tally input batch, consisting of the duo $B^X{}_k = R^X$, $D^Y = D^Y{}_k$, which can be written as B_k, D_k. A series of k mix operations[1] transforms each such duo into a corresponding ballot image B^Z. The lth mix transforms each duo B_l, D_l in its input batch into a corresponding B_{l-1}, D_{l-1} duo in its lexicographically ordered output batch by decrypting D_l using its secret decryption key corresponding to e_l, extracting d_l' from the resulting plaintext, applying h', and finally applying $B_{l-1} = d_l \oplus B_l$. The kth mix performs the same operation on each duo, and because D_0 is empty, the result is $B_0 = B^Z$.

Prior arrangement partitions the k mixes into contiguous sequences of four among a set of $k/4$ trustees, where k is divisible by four. For simplicity, assume that the input batch size is also divisible by four. When the mixing is complete, half the tuples in each batch are selected for opening. The work of Markus Jakobsson, Ari Juels, and Ronald Rivest[2] inspired this approach. A random public draw, such as that used for state lotteries, ensures that these choices are independent and uniformly distributed. The tuples selected for opening depend on the order in each trustee's four mixes:

- In the first mix, half of all tuples are opened.
- In the second, the tuples not pointed to by those opened in the first mix are opened.
- In the third, half the tuples pointed to by those opened in the second mix and half the tuples not pointed to are opened.
- For the fourth mix, as with the second, those tuples not pointed to by the previous mix are opened.

A few extensions are worth noting at this point. For improved privacy, multiple doll pairs allow separate ballot images per contest and/or question. Error correction can be provided in the space around the votes. Also, to prevent a voter's choice of layer, which is revealed to the poll workers, from determining the ballot image type, and to prevent bias in voter preference for particular layers, the dolls can determine a mapping between the physical layers and a pair of symbols that the voter chooses between. The symbols are printed before layer selection in a way that hides them until after the layers are separated.

References

1. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Comm. ACM*, vol. 24, no. 2, 1981, pp. 84–88.
2. M. Jakobsson, A. Juels, and R.L. Rivest, "Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking," *Proc. Usenix Security 2002*, Usenix Assoc. 2002, pp. 339–353; also available as IACR reprint 2002/025.

ing power. The receipt system uses such unconditionally secure techniques to ensure that integrity is not compromised except with the probabilities of detection enforced.

Most cryptography used in practice, however, is computationally secure—that is, in principle it is breakable if enough computing power is applied. No criminal has likely been able to make such computations using resources available today (because many systems, including international high-value wire transfer, that rely on such codes are still in place). Such standard cryptographic building blocks, which are also like those used widely by browsers when accessing secure Web sites, are enough (along with addition modulo two) to build the systems described here.

The receipt system uses computationally secure encryption to form the layers, which ultimately encrypt the data in receipts and batches, and thus protect privacy and ballot secrecy. After voting, the codes protecting receipts and posted batches, which are only readily linkable to ballot numbers and not people (apart from perhaps the case of provisional ballots), can easily be as good as those protecting comparable and much more identifiable, sensitive, and detailed data traveling on networks today.

Technical provision of privacy in voting is limited, however. Because of current surveillance technology, such as sensors like miniature cameras and emanation receivers, as well as memory and transmitters, the confidentiality of what transpires in voting booths cannot in practice be held to any absolute standard. Other limiting factors include

- Most US voter party affiliations are a matter of public record.
- The more a device helps a voter the harder it is to keep it from learning who they vote for (although, as in the system proposed here, devices need not be able to retain data between votes).
- Even the "gold standard" of voting systems—manual paper ballots—is subject to marking or ballot number recording and automatically captures fingerprints.
- Theoretical limits generally force a choice in cryptographic systems between unconditional integrity and unconditional privacy.

Thus the system presented here is arguably optimal. It protects privacy computationally according to current best practices by encrypting votes in receipts and published batches. And it protects the tally's integrity unconditionally by enforcing sufficient probabilities of detecting tampering.

Proof sketches

The properties asserted informally in the text can be abstracted and stated more precisely in terms of the more formal description provided (in the sidebar on page 44). Without implying any particular level of rigor, explanations for these statements can be illustrated in terms of the familiar format of theorems and proof sketches.

Theorem 1

If, for a selected and an unselected 4-tuple from an instance of step 2 in the voting process, the selected 4-tuple satisfies the consistency check in step 6 and there is a 2-tuple that would satisfy such a check with the unselected 4-tuple, the doll of the unselected layer, as printed on the selected layer, is correctly formed and determines all white pixels printed on the unselected layer (relative to which the voter sees the vote in the receipt's red bits).

Proof (sketch): The serial number q and the doll D^Y are printed on both layers identically, as the voter verifies in step 3. The doll D^Y in the unselected layer's 2-tuple is correctly determined by q, according to the functions s^Y, h, and e, because the unselected 4-tuple would satisfy the consistency check in the hypothetical step 6. Similarly, q correctly determines the white bits W^Y according to s^Y, h, and h' that the voter checks in the hypothetical step 6 as being correctly printed on the unselected layer. Because the encryption e is bijective, D^Y determines the d'^Y_i, which determines W^Y. Thus, the D^Y printed on the receipt determines the W^Y printed on the unselected layer.

Theorem 2

Any properly formed, selected layer and its resulting processing reveal the ballot images only in encrypted form until they appear in the tally batch.

Proof (sketch): Of the selected layer's six components $<L^X, q, D^t, D^b, s^X(q), o^X(L^X, q, D^t, D^b, s^Z(q))>$, only the first depends on the ballot image B. The L^X bits are partitioned among the R^X bits that depend on B, and the W^X that don't. The W_i^X are each encrypted by e_i and can therefore be ignored. Each B_l, $1 \le l \le k$, appears in its respective input batch summed modulo 2 with each d_p, $l \le p < k$. Thus, each time any B appears in an input batch it appears ⊕ed with a distinct pseudorandom value that only appears in all following sums. The resulting set of linear equations thus cannot be solved for any B.

Theorem 3

For any trustee's mixes, a duo's prescribed opening doesn't reveal a restriction on the correspondence between any individual input and output.

Proof (sketch): It's easy to see that the restriction imposed by an odd-numbered batch followed by an even-numbered batch—a *doubleton* of batches—requires that each of the two known halves of the inputs results in a respective known half of the outputs. (Note, however, that this could reveal something about an individual input and output, such as whether the input could correspond to a particular unique output.) A next doubleton that exactly splits each output partition of its predecessor across its own input partitions enforces the restriction that exactly half the members of an input partition are in each output partition, but leaves any particular input to the two doubletons free to be any particular output.

Theorem 4

The probability that a trustee that improperly forms u distinct duos in any of its output batches will be detected in at least one duo is $1 - 2^{-u}$.

Proof (sketch): The random draw selects the duos to be opened in a trustee's first batch independently of the trustee's control; an opened duo is either correct or not. The probability of detection is thus 50 percent for each improperly formed duo in the batch. Because the opened values are all correct, the half chosen for the next batch is selected independent of any improperly formed duo, and so on inductively.

This new type of receipt system reduces the cost of integrity while raising its level dramatically and making its assurance open to all interested parties. Robustness is similarly more cost-effective and raised to a level where it too can be ensured by voters (assuming they can access a functioning booth) through their receipts. Privacy and secret-ballot protections can easily meet current best practices and are arguably practically optimal. Improved functionality of the system facilitates accessibility and higher turnout, as well as needed improvements in adjudication. Perhaps most fundamentally, it can do a great deal to repair and improve voter confidence.

The hardware costs of these systems can be lower than current black box systems, which governments buy at many times the price of open-platform PCs. The cost of suitable printers in volume should be considerably less than the hardware cost saving. This doesn't even include savings in maintenance, upgrade flexibility, multiple uses, and reductions in outmoded security provisions. In fact, because of the provable integrity, federal dollars could be very well spent sponsoring development of such systems and making them available.

The Help America Vote Act is funding the introduction of computers into almost all voting booths in the US over the next few years, and the systems that are deployed through this unprecedented funding will likely be in place for a long time. (There is also, for instance, an effort to automate Latin-American voting using the Brazilian model, which also includes computers in voting booths.) A growing grassroots movement is pushing to allow voters to see a printed summary of their vote, which is retained for possible recount. So far such sum-

maries have not been shown to be effective or workable in general and have been replaced in Brazil. This movement does, however, indicate a growing level of public concern, and the two printing approaches could even be combined. The sad truth, however, is that the process of deciding which types of systems to deploy has so far for the most part been closed and informed neither by explicit performance requirements nor generally accepted security practices.

The receipt system presented here offers a new level of integrity, access, robustness, and adjudication, all at lower cost, that make it a compelling way to secure polling-place elections—and it should be the only way acceptable now. □

Acknowledgments

It is a pleasure to acknowledge Ron Rivest, who served as a superb sounding board for ideas. The "WOTE" workshop was also very stimulating. Later, Jim Dolbear and Lori Weinstein provided a lot of help. Detailed comments from Josh Benaloh, Paul Craft, David Jefferson, Doug Jones, and Andreu Riera, as well as feedback from Jeremy Bryans, Dan Boneh's group, Stuart Haber, Robert Naegele, Peter Ryan, Marius Schilder, and Adi Shamir were also helpful.

References

1. M. Naor and A. Shamir, "Visual Cryptography," *Proc. Advances in Cryptology* (*Eurocrypt 94*), A. De Santis, ed., LNCS 950, Springer-Verlag, 1995, pp. 1–12.
2. C.E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical J.*, no. 28, 1949, pp. 656–715.

David Chaum *is currently affiliated with several companies, universities, and international projects. Widely recognized as the inventor of electronic cash, he also originated a number of basic cryptographic techniques, general results, and techniques that allow individuals to protect their identity and related information in interactions with organizations. He has more than 50 original technical publications and 25 separate cryptography-related patent filings. Chaum has a PhD in computer science from the University of California, Berkeley. He has taught, led a crypto research group, and founded DigiCash and the International Association for Cryptologic Research (IACR). Contact him at info@chaum.com.*