CS 526 Information Security: Assignment 5

John Ross Wallrabenstein (jwallrab)

October 8, 2010

1 Problem 1

Before we address the question of which properties the proposed signature schemes lack, we enumerate desirable properties of any digital signature system:

1. **Authenticity**: A digital signature should be *authentic*, in that the signature will convince the recipient that the claimed author deliberately signed the document. It should be computationally infeasible for a third party to forge the author's signature.

2. **Integrity**: A digital signature on a document should verify as correct if and only if the document signed has not been altered after the signature was applied.

3. **Unique**: A digital signature on a document should be *unique*, in that the signature is only valid for the signed document and cannot be transferred to a different document.

4. **Nonrepudiation**: A digital signature on a document cannot be repudiated by the signer. That is, a digital signature of a document should have non-trivial legal value. It must be difficult to prove beyond a reasonable doubt that the signature was produced by a party other than the signer in a court of law.

5. **Efficiency**: A digital signature should be efficiently computable. That is, producing signatures for large documents should be minimally computationally expensive.

6. **Compact Representation**: A digital signature should have a compact fixed representation, independent of the length of the document signed.

7. **Offline Verification**: A digital signature should be verifiable without requiring interaction with the signer. That is, given a signed document, the recipient can verify the signature without communication with the signer.

Goldwasser et. al. defined the fundamental notions of digital signature scheme security [4].

1.1

Foreword: The issue of Public Key Infrastructure (PKI) is non-trivial, and well beyond the scope of this assignment. Therefore, we assume that no certificate authorities were involved in any of the protocols. Thus, both the problem's signature formulation, as well as the proposed modifications, are vulnerable to a Man-In-The-Middle or Bucket Brigade attack by an adversary.

1. **Offline Verification**: The use of a public key system in the problem allows signatures to be verified while the signer is offline. That is, the recipient can obtain the signer's public key from any public key server to decrypt the message digest to verify the document signature. The proposed modification requires that the signer participate in the verification protocol by decrypting the received signature and returning the result

of the verification to the receiver. However, this concern is dwarfed by the next:

2. **Authenticity**: The proposed modifications make it impossible to verify the authenticity of a message. The sender is the only entity that can verify their own signature, which in essence implies that anyone could masquerade as the sender. Assume an adversary $\mathcal{A}$ wishes to masquerade as Alice to Bob. $\mathcal{A}$ generates a message m addressed to Bob from Alice, and uses $E_{\mathcal{A}}(H(m))$ as the signature. Bob receives the pair $\langle m, E_{\mathcal{A}}(H(m))\rangle$ and sends $E_{\mathcal{A}}(H(m))$ to $\mathcal{A}$ to check the signature. $\mathcal{A}$ returns that the signature is valid, and Bob accepts that the message was authored by Alice. As the symmetric key $k_{\mathcal{A}}$ is known only to $\mathcal{A}$, Bob cannot distinguish between an encrypted hash generated by Alice with k_{Alice} or one generated by the adversary with $k_{\mathcal{A}}$.

1.2

1. **Efficiency**: Encrypting the entire message with the sender's private key requires computation on the order of that required to provide confidentiality, without actually providing such a service. That is, anyone can decrypt the message with the sender's public key, so confidentiality is not provided. However, the same number of modular exponentiations required to provide confidentiality are required under this modification to provide authenticity alone. Modular exponentiation, required for (all common) public key operations, are orders of magnitude more computationally expensive than similar symmetric key and message digest operations (usually bitwise operations). The hashing method suggested in the problem description requires that only the message digest be encrypted, which would perhaps require a single modular exponentiation. That is, a standard message digest is less than 2^{512}, while a public key modulus is usually in excess of 2^{2048}.The proposed modification would require $O(b)$ modular exponentiations, where b is the number of bits in the message m. The hashing method requires $O(1)$ modular exponentiations, when the hashing function is fixed.

2. **Compact Representation**: One could argue that this scheme (non-hashing) combines the message and signature to create a very compact representation, or that because the message *is* the signature, it is not compact. In either case, this concern is dwarfed by the loss of efficiency.

3. **Domain**: The message m to be signed may be chosen from an arbitrary domain; one, perhaps, that is not a subset of the domain of the signature scheme. Public key cryptosystems (usually) operate on numbers modulo a security parameter n, and hash functions will output a value $H(m)$ in this domain. Thus, it is natural to use a hash function to transform the domain of the message $m \in \{0,1\}^*$ to the domain of the public key system used to sign the message, usually $\mathbb{Z}_n$.

2 Problem 2

2.1

We begin by enumerating desirable properties of an electronic voting system:

1. Only registered voters can vote. [7, 8, 3]
2. No person can vote more than once. [7, 8, 3]
3. No one can determine for whom anyone else voted. [7, 8, 3]
4. Every voter can make sure that his vote has been counted. [7, 8]
5. No person can duplicate any other person's vote. [7, 8]
6. No person can change any other person's vote undetected. [7, 8]

7. All valid votes are counted correctly. [3]
8. The dishonest voter cannot disrupt the voting. [3]
9. Nothing must affect the voting (Fairness). [3]
10. *Opt.* Everyone knows who voted, and who did not. [7]
11. *Opt.* A voter can change his mind within a given time period. [7, 8]
12. *Opt.* If a vote is miscounted, the voter can identify and correct the problem without jeopardizing the secrecy of his ballot. [7, 8]

Designing such a system to support these properties is non-trivial. Further, ensuring that adversaries cannot behave in a manner prevented by current voting systems is a more difficult problem. Most protocols use one or more Central Tabulating Facilities (CTF), which in practice would be government servers. While protocols exist that do not require a CTF, they are likely infeasible for large-scale presidential elections. In any case, some trusted party must be responsible for handling the authentication of voters. Consider the following issues that arise when addressing such concerns:

1. **Masquerading**: An adversary may acquire the voting credentials of persons who have no interest in voting. If the legitimate voters are unaware of, or indifferent to the actions of the adversary, then the electronic voting system allows actions prevented by current systems. Of course, current systems do not protect against bribing voters (although the adversary cannot guarantee the voter(s) fulfilled the contract). For a solution that prevents adversaries from bribing voters, see [6].
2. **Privacy**: Preserving the privacy of voters (w.r.t. the CTF) is particularly difficult, as the CTF usually can associate an individual voter with their identification number. While the voter may have privacy with respect to the general population, this is not necessarily the case with respect to the CTF (government). The protocol for authentication we describe preserves the privacy of the voter's ballot with respect to both the CTF and the general population.
3. **Assigning IDs**: Before the election, some trusted party (e.g. CTF, Government) must associate some identification information, such as a unique identification number, with a registered voter. Addressing duplicates is handled by the protocol, so we omit this problem. However, the assignment of IDs to voters should require a process similar to distributing social security cards.
4. **Disputes**: Two potential dispute cases seem salient: voters masquerading as another legitimate voter, and a voter who claims their vote was not counted, or does not match their submission. We have described the former, so we address only the latter. Many voting systems do not provide protocols for a voter to prove that the CTF modified or omitted their vote, or that they are the rightful person associated with the identification number.

2.2

We present the voting protocol of Fujioka et. al. [3], primarily because the authentication procedure was formally verified by Mahrooghi et. al. [5]. Blind signatures are used as a tool when authenticating whether or not a voter has the right to vote [2]. Alternatively, the All-Or-Nothing Disclosure of Secrets method proposed by Brassard et. al. may be substituted [1]. As the question asks us to address the authentication procedure only, we omit the full details of the voting protocol.
Assume the following notations:

- V_i: Voter i
- A: Administrator (CTF)

- $\zeta(v, k)$: Bit-commitment scheme for message v using key k
- $\sigma_i(m)$: Voter V_i's signature scheme
- $\sigma_A(m)$: Administrator's signature scheme
- $\chi_A(m, r)$: Blinding technique for message m and random number r
- $\gamma_A(s, r)$: Retrieving technique of blind signature
- ID_i: Voter V_i's identification
- v_i: Vote of voter V_i

1. Voter V_i selects vote v_i and completes the ballot $x_i = \zeta(v_i, k_i)$ using a key k_i randomly chosen.
2. V_i computes the message e_i using blinding technique $e_i = \chi(x_i, r_i)$.
3. V_i signs $s_i = \sigma_i(e_i)$ to e_i and sends $\langle ID_i, e_i, s_i \rangle$ to A.
4. A checks that the voter V_i has the right to vote. If V_i doesn't have the right, A rejects V_i.
5. A checks the signature s_i of message e_i. If they are valid, then A signs $d_i = \sigma_A(e_i)$ to e_i and sends d_i as A's certificate to V_i.
6. A announces the number of voters who were given its signature, and publishes a list that contains $\langle ID_i, e_i, s_i \rangle$.

Of course, verifying that V_i has the right to vote is no trivial matter. In a true implementation, it would likely require that a trusted party (e.g. notary, election official) verify that a voter matches their state identification or driver's license, and that they are registered to vote in the district. If the voter meets these requirements, then the trusted party would sign the voter's (hidden, but valid) vote.

2.3

*Foreword: We take this question to refer to authentication and identity management **in general**, rather than authentication and identity management with respect to electronic communication.*

No - in the sense that cryptography is decidedly **not** necessary for authentication and identity management in many real world protocols. For example, we are able to distinguish people based on physical appearance, voice tone, personality and speech patterns. Thus, many day-to-day protocols (e.g. calling friends/family) do not require cryptography for authentication. Similarly, n-factor authentication is commonly used to verify the identity of a person you do not personally know. For example, producing a photo identification (State ID Card, Driver's License) and a personal document (Birth Certificate, Social Security Card) that match is an example of 2-factor authentication that does not require cryptography.

3 Problem 3

3.1

3.1.1

Separation of Privilege vs. Economy of Mechanism: The principle of separation of privilege is likely to conflict with the economy of mechanism principle when the design application is for personal, rather than business, systems. That is, separation of privilege may take precedence in a corporate setting where the potential to damage thousands of clients exists. However, in a personal setting, the economy of mechanism takes precedence given the reduced possibility of widespread damage.

3.1.2

Complete Mediation vs. Psychological Acceptability: The principle of complete mediation is likely to conflict with the psychological

acceptability principle when large batch operations are involved. That is, if a user wishes to perform a batch operation, requiring complete mediation would make the task tedious and cumbersome for the user. In this instance, psychological acceptability would take precedence.

3.2

We consider the scenario where the principle of separation of privilege conflicts with the principle of economy of mechanism.

Certificate Authorities

Consider a certificate authority CA that signs certificates after verifying both the individual I's identity in person, and their rightful ownership of a particular domain D. The certificate authority is trusted by all major computer manufacturers, and their public key is loaded onto systems before they are distributed for sale. If an adversary $\mathcal{A}$ could successfully gain a signature from CA on a domain they do not administer, $\mathcal{A}$ could masquerade as the true domain D, and users would not receive a warning from their browser. This has the potential to affect a large number of people, particularly when the site in question handles banking or other financial services.
The tradeoff involves requiring individuals wishing to obtain a signature to appear in person, in exchange for the greater security afforded by such a measure. Rather than use a single factor for authentication, such as a social security number, multiple factors are required to obtain a signature. The individual is usually required to produce both a photo identification card (driver's license, passport) and a personal document (original birth certificate, social security card) in order to verify their identity. Further, the individual must possess documentation of their ownership of the domain D. The entire transaction must take place in person, as signatures distributed to individuals who do not administer domains will have far reaching consequences. Thus, separation of privilege (requiring multiple conditions) takes precedence over economy of mechanism due to the implications a failure in the procedure would have.

Personal Computing

Consider an individual I who owns a personal computing system S. In order to access the system S, I is required to provide a valid login credential C. It is assumed by S that any entity possessing a valid credential C should be given access to the system, with no further conditions necessary for approval. In this scenario, the principle of economy of mechanism takes precedence over the principle of separation of privilege. That is, for a personal computing system S, requiring 2^+-factor authentication would be overly complicated and cumbersome for legitimate users wishing to access S. However, the principle of separation of privilege would require that at least one other factor (other than C) be required in order for access to S to be granted. In the extreme case, this would require the approval of a separate individual $I^{`}$, rather than a second authentication factor (such as a SecurID card or biometrics). Given that the potential for damage is limited (compared with a certificate authority) for personal computing systems, this trade-off is justified. That is, a compromised personal computing system is likely to damage a small number of users, whereas a compromised certificate authority could have serious ramifications for millions of users. Users would be reluctant to adopt systems that required multiple authentication protocols to gain access, so the trade-off provides acceptable security while not requiring an unnecessarily complicated security mechanism to provide it.

4 Problem 4

4.1

4.1.1

Assumption: Given that receiver/sender are always referred to with singular nouns, we assume that only a single receiver and a single sender

are present in the scenario. By definition, this is a noiseless channel. That is, a noiseless channel is a covert channel that uses a resource (data diode) available only to the sender and receiver. As there are only a single sender and receiver in the problem definition, they are the only entities with access to the data diode.

4.1.2

To calculate the capacity of this covert channel, one would need to know:

- t: The time required for the sender to fill the buffer entirely. It could be reasonably assumed that the receiver could empty the buffer in the same amount of time.
- c: The capacity, or size, of the buffer.

In this covert channel, a bit is transmitted based on whether or not, upon sending $c+1$ bits to the buffer, an acknowledgement is received by the sender. If the receiver wishes to transmit a 0, they do not send any acknowledgements to the data diode, causing the last packet sent by the sender to not receive an acknowledgement (it was dropped by the data diode). To transmit a 1, the receiver acknowledges at least one packet so that the final packet (containing $c+1$) does not overflow the buffer, and hence receives an acknowledgement.

4.1.3

As the channel is noiseless, the capacity is a function of the time t required to send $c+1$ bits to (potentially) overflow the buffer if the receiver does not acknowledge any packets. Thus, we have that:

$$capacity = \frac{1}{t} bits/second$$

For example, if it takes 0.5 seconds to fill the buffer, where each time the buffer is filled transmits a single bit, then the capacity of the channel is:

$$capacity = \frac{1}{t} = \frac{1}{\frac{1}{2}} = 2 bits/second$$

4.2

4.2.1

A potential *timing attack* exists with respect to the computational power of the data diode. If the data diode is capable of only sending an acknowledgement to the sender or receiving an acknowledgement from the receiver, a covert channel exists. That is, to transmit a bit b the receiver could acknowledge packets immediately, creating a pause between acknowledgements sent to the sender. Otherwise, the receiver could transmit $1-b$ by waiting to acknowledge a packet to the data diode. Thus, the data diode must ensure that all acknowledgements sent to the sender are spaced evenly.

4.2.2

Even if the data diode attempts to space acknowledgements to the receiver evenly, another *timing attack* exists. If the receiver wishes to transmit a bit b, it can flood the data diode with acknowledgements (which may or may not be for valid sequence numbers). This will overwhelm the data diode, and cause a pause between acknowledgements sent to the sender. Similarly, to transmit a bit $1-b$, the receiver could respond with acknowledgements as usual, allowing an uninterrupted stream of acknowledgements to be sent to the sender. This channel requires that the sender and receiver agree on some interval, so that the sender can distinguish between multiple identical bits sent in sequence. Finally, note that this attack does *not* require the acknowledgements sent to the sender to be synchronized with acknowledgements sent by the receiver. Even if the data diode attempts to respond immediately (as in the problem) to the sender, it can be overwhelmed with acknowledgements by the receiver.

References

[1] Brassard, G., Crépeau, C., and Robert, J.-M. All-or-nothing disclosure of secrets. In *Proceedings on Advances in cryptology—CRYPTO '86* (London, UK, 1987), Springer-Verlag, pp. 234–238.

[2] Chaum, D. Blind signatures for untraceable payments. *Advances in Cryptology* (1983), 199–203.

[3] Fujioka, A., Okamoto, T., and Ohta, K. A practical secret voting scheme for large scale elections. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology* (London, UK, 1993), ASIACRYPT '92, Springer-Verlag, pp. 244–251.

[4] Goldwasser, S., Micali, S., and Rivest, R. L. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput. 17* (April 1988), 281–308.

[5] Mahrooghi, H., Haghighat, M., and Jalili, R. Formal verification of authentication-type properties of an electronic voting protocol using mcrl2. In *Fourth International Workshop on Verification and Evaluation of Computer and Communication Systems* (2010), VECoS.

[6] Niemi, V., and Renvall, A. How to prevent buying of votes in computer elections. In *Advances in Cryptology ASIACRYPT'94*, J. Pieprzyk and R. Safavi-Naini, Eds., vol. 917 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1995, pp. 164–170.

[7] Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*, 2nd ed. Wiley, October 1996.

[8] Wagstaff, Samuel S, J. *Cryptanalysis of Number Theoretic Ciphers*, first ed. Computational Mathematics Series. Chapman & Hall/CRC Press, 2003.