12. Exercise sheet
Hand in solutions until Sunday, 18 July 2010, 23.59 h

We take a look at the remaining seven protocols not discussed so far: REENCpF, VOTEpF, PET, MIXNET, REGISTER, VOTE, TABULATE (see the appendix). The aim of this assignment is to get a first hands-on experience with them.

Exercise 12.1 (VOTEpF and MIXNET). (6+6 points)
Consider VOTEpF and MIXNET. What is the purpose of these protocols? Answer with a complete English sentence without mathematical symbols. Also, state the information that is verified in each case. Discuss further important properties.

Exercise 12.2. (7+7 points)
Answer the following questions for the named protocol.

REENCpF What happens if a lazy prover chooses the random value $t_i = s_i$ in step 3.

VOTEpF What are similarities and differences to KNOWDLOG?

PET What are similarities and differences to EQDLOG?

MIXNET What is the purpose of the $q_i$?

REGISTER Why the use of nonces instead of simple random choices?

VOTE Why is this called protocol, not algorithm?

TABULATE Concerning the chronological sequence of steps 4-11, which can be run in parallel, for which can the order be reversed?

Exercise 12.3. (0+4 points)
Assume a scenario where $n$ Tabulation tellers and $m$ voters are involved. How often is every protocol executed (on average/at least/at most)?
A. Appendix

Public input: A list $C = [(T_i, Y_i)]_i$ of (reencrypted) ciphertexts, a particular ciphertext $\tilde{C} = (T, Y)$, and the recipients' public key $X$.
Private input to the prover: An index $j$ into the list $C$ and the reencryption randomness $t'$ such that $\tilde{C} = C_j + \text{enc}_X(O; t')$.
Output to the prover: REENCPf$(j, t') = (s, \tilde{t})$

1. The prover performs 2–8.
2. For all indices $i$ of $C$ do 3–5
3. She picks random values $s_i, t_i \in \mathbb{Z}_q$.
4. $\tilde{T}_i = s_i(T_i - T) + t_iP$ and
5. $\tilde{Y}_i = s_i(Y_i - Y) + t_iX$.
6. The prover computes $c \leftarrow \mathbb{Z}_q(\text{hash}(\tilde{C}, C, [(\tilde{T}_i, \tilde{Y}_i)]_i))$.
7. The prover computes $\tilde{s}_j \leftarrow c - i \neq j s_i$, and for $i \neq j$ let $\tilde{s}_i \leftarrow s_i$, $t_j \leftarrow t_j - t'(\tilde{s}_j - \tilde{s}_j)$, and for $i \neq j$ let $\tilde{t}_i \leftarrow t_i$.
8. He sends $(s, \tilde{t})$.
10. He reconstructs $\tilde{T}$ and $\tilde{Y}$:
11. For all indices $i$ of $C$ do 12–13
12. $\tilde{T}'_i = \tilde{s}_i(T_i - T) + \tilde{t}_iP$ and
13. $\tilde{Y}'_i = \tilde{s}_i(Y_i - Y) + \tilde{t}_iX$.
14. He computes $c' \leftarrow \mathbb{Z}_q(\text{hash}(\tilde{C}, C, [(\tilde{T}'_i, \tilde{Y}'_i)]_i))$, and $d' \leftarrow \sum_i \tilde{s}_i$.
15. He verifies $c' \equiv d'$.

Public input: Encrypted credential $(T_1, Y_1, c, r) = \text{CredEnc}(s, t, K_{TT}, rid, vid)$, encrypted choice $(T_2, Y_2)$, the prover’s public key $X$.
Private input to the prover: Temporary keys $t_1, t_2 \in \mathbb{Z}_q$ such that $T_1 = t_1P$.

1. The prover picks $s_1, s_2 \in \mathbb{Z}_q$.
2. The prover computes $c \leftarrow \mathbb{Z}_q(\text{hash}(P, X, T_1, Y_1, T_2, Y_2, s_1P, s_2P))$.
3. The prover computes $r_1 \leftarrow s_1 - c t_1$ in $\mathbb{Z}_q$.
4. He sends $(c, r_1, r_2)$.
5. The verifier checks $c \leftarrow \mathbb{Z}_q(\text{hash}(P, X, T_1, Y_1, T_2, Y_2, r_1P + cT_1, r_2P + cT_2))$.

Public input: Two ciphertexts $C_j = (T_j, Y_j)$, encrypted with the tabulation tellers’ common public key $X_{TT} = \sum_i X_i$. 
Private input to tabulation teller $i$: The private key share $x_i$.

Output: PET($C_1$, $C_2$)

1. Tabulation teller $i$ performs 2–6.

2. Pick a randomizer $z_i \in \mathbb{Z}_q$ and compute
   \[
   \tilde{T}_i = z_i(T_1 - T_2), \quad \tilde{Y}_i = z_i(Y_1 - Y_2).
   \]
   \[\text{commit}(\tilde{T}_i, \tilde{Y}_i)\]

3. Publish a commitment to $(\tilde{T}_i, \tilde{Y}_i)$.
4. Wait until commitments of all tabulation tellers are available.
5. Publish $(\tilde{T}_i, \tilde{Y}_i)$ and a proof of equality of discrete logarithms for $(T, Y, \tilde{T}_i, \tilde{Y}_i)$.
   \[\text{(}\tilde{T}_i, \tilde{Y}_i, \text{EqDlogs(\ldots)}\text{)}\]
6. Wait and verify all commitments and proofs.
7. Let $\tilde{T} = \sum_i \tilde{T}_i, \quad \tilde{Y} = \sum_i \tilde{Y}_i$.
8. All tabulation tellers jointly decrypt $(\tilde{T}, \tilde{Y})$:
   \[m' \leftarrow \text{DistDec}(\tilde{T}, \tilde{Y}).\]
9. If $m' = \mathcal{O}$ then Return Equal Else Return Unequal.

**Algorithm A.4.** Atomic mix operation (MIX).

Input: A list $C = [C_i]_i$ of ciphertexts, and a direction $d \in \{\text{In}, \text{Out}\}$.

Output: An anonymized reencryption $M = \text{Mix}(C)$ of $C$, and a list of commitments.

Private output: $r, w, p$.

1. Pick a permutation $\pi$ of the indices of $C$. (Instead of picking it, you can also compute it such that the reencrypted list $M$ is sorted.)
2. If $d = \text{In}$ then $p \leftarrow \pi^{-1}$ Else $p \leftarrow \pi$.
3. Pick reencryption randomnesses $r_i \leftarrow \mathbb{Z}_q$ and commitment randomizers $w_i \leftarrow \mathcal{R}$.
4. Let $M \leftarrow [\text{Reenc}(C_{\pi(i)}; r_i)]_i$.
5. Let $S \leftarrow [\text{Commit}(w_i, p(i))]$.

**Protocol A.5.** The anonymizing mix net (MIXNET).

Public input: A list $C = [C_i]_i$ of ciphertexts.

Output: Anonymization $\text{MIXNET}(C)$ of $C$.

1. Let $M_{0,2} \leftarrow C$.
2. For $i = 1 \ldots n$ do 3–6
3. Wait for $M_{i-1,2}$.
4. Mix $i$ computes $(M_{i,1}, S_{i,1}) \leftarrow \text{Mix}(M_{i-1,2}, \text{Out})$ and publishes that.
5. Mix \(i\) computes \((M_{i,2}, S_{i,2}) \leftarrow \text{Mix}(M_{i,1}, \ln)\) and publishes that.

6. Pick a further random value \(q_i \leftarrow R\) and publish a commitment to it.

7. Wait for all mixes to finish.

8. Then each mix publishes \(q_i\).

9. Wait and verify all other mixes’ commitments.

10. Let \(q \leftarrow \text{hash}(q_1, \ldots, q_n)\).

11. Compute the challenge \(c_i \leftarrow \text{hash}(q, i)\).

12. For \(i \in \{1, \ldots, n\}\) in parallel do 13–20

13. Mix \(i\) publishes \(r_j\) or \(r_{p(j)}\) depending on \(\text{bit}_j(c_i)\), \(w_j\) and \(p(j)\) from the mixing resulting in \(M_{i,1+\text{bit}_j(c_i)}\) for all indices \(j\) of \(C\).

14. Now all the mixing information can be erased.

15. Wait for the other mixes’ responses.

16. Verify Commit(\(w_j, p(j)) = S_{i,1+\text{bit}_j(c_i)}\).

17. If \(\text{bit}_j(c_i) = 0\) then

18. Verify Reenc\(_X(M_{i-1,2,p(j)}; r_j) = M_{i,1,j}\).

19. Else

20. Verify Reenc\(_X(M_{i,1,j}; r_{p(j)}) = M_{i,2,p(j)}\).

21. Return \(M_{n,2}\)

**Protocol A.6. Registration (REGISTER).**

Public input: The distributed public key \(X_{TT}\) of the tabulation tellers, a public RSA key \(K_{RT_i}\) of the registration teller \(i\). The voter’s public designation key \(X_{vid}\). The voter’s public registration RSA key \(K_{vid}\). Identifiers of election (eid), voter (vid), registration tellers (rid), and block (bid). Public credentials \(S_j = \text{CredEnc}(s_j; r; X_{TT}; \text{rid}, \text{vid})\) for each registration teller \(j \in \text{rid}\).

Private input to registration teller \(RT_i\): Private credential \(s_i \in \mathcal{M}\) and encryption randomness \(r \in \mathbb{Z}_q^\times\).

Private input to the voter: Private registration RSA key \(k_{vid}\), \ldots

Output to the voter: private credentials \(\text{Register}(\text{vid}, \text{rid}, \text{sid})\)

1. The voter picks a nonce \(N_{vid}\) and sends the election id eid, his id vid, and the nonce encrypted to the registration teller \(i\).

2. The registration teller \(RT_i\) verifies that vid is a voter in block (precinct) bid in election eid, and that for each registration tellers \(j\) in
rid the public credential \( S_j \) is available and \( \text{CredVer}(S_j; j, \text{vid}) \) succeeds.

3. The registration teller picks a nonce \( N_R \) and an AES key \( k \) (of security level \( \ell \)).

4. Send the registration teller ids rid, the nonces \( N_R \) and \( N_V \) and the chosen AES key \( k \) to the voter.

5. The voter decrypts and verifies rid and \( N_V \), and sends the nonce \( N_R \) back to the registration teller RT.

6. The registration teller \( RT_i \) verifies \( N_R \).

7. The registration teller picks \( r' \leftarrow Z_q^* \) and computes \( w \leftarrow r' - r \) and another encryption \( S'_i \leftarrow \text{Enc}(s_i; r', X_{TT}) \) of the private credential.

8. The registration teller sends AES encrypted the private credential share and the new randomness \( r' \) together with a designated verifier proof that \( S_i \) and \( S'_i \) encrypt the same message.

9. The voter decrypts and verifies the designated verifier proof against \( S_i \) from the bulletin board.

\[
\text{RSAenc}_{K, \text{vid}}(\text{rid}, N_R, N_V, k) \\
\begin{array}{c}
\downarrow N_R \\
\end{array}
\]

\[
\text{AESenc}_k(s_i, r', \text{DVRP}(\ldots), \text{bid}) \\
\begin{array}{c}
\downarrow \text{AESenc}_k(s_i, r', \text{DVRP}(\ldots), \text{bid})
\end{array}
\]

**Algorithm A.7.** Fake credentials (\textsc{FaceCredential}).

Input obtained from registration: Private credential shares \( s_i \), public credential shares \( S_i \), reencryption factors \( r_i \), and designated verifier proofs \( D_i \) from each registration teller \( RT_i \).

Input: Index set \( L \) of registration teller for which to fake shares. The voter’s designation key pair \( (X_{\text{vid}}, x_{\text{vid}}) \).

Output: Fake private credential shares . . .

1. For \( i \) do 2–9
2. If \( i \in L \) then
3. Pick \( \tilde{r}_i \leftarrow Z_q^* \).
4. Pick \( \tilde{s}_i \) randomly.
5. Else
6. Let \( \tilde{r}_i \leftarrow r_i \).
7. Let \( \tilde{s}_i \leftarrow s_i \).
8. \( \tilde{S}_i \leftarrow \text{enc}(\tilde{s}_i; \tilde{r}_i; X_{TT}) \).
9. Compute a non-interactive fake designated verifier proof \( \tilde{D}_i \) by Protocol 6.4
10. Return \( [(\tilde{s}_i, \tilde{r}_i; \tilde{D}_i)]_i \)
Protocol A.8. Vote (VOTE).
Public input: The distributed public key $X_{TT}$ of the tabulation tellers. Well-known choice ciphertext list $C$.
Private input: The voter’s choice $t$ and his credentials $s$.
Output to the ballot box: Vote$(t, s)$
1. The voter picks a randomness $r_s$ and encrypts his credentials $S \leftarrow \text{enc}(s; r_s; X_{TT})$ for the tabulation tellers.
2. He picks a randomness $r_v$ and reencrypts his choice $C_t; V \leftarrow \text{reenc}(C_t; r_v)$.
3. He prepares a vote proof $P_w$ of correct voting by Protocol A.2 with inputs $S, V, r_s, r_v$, and further context.
4. He prepares a $\text{RENCPF} P_k$ that $V$ is a reencryption of one of the ciphertexts $C$ by Protocol A.1.
5. Let vote $\leftarrow (S, V, P_w, P_k)$ and send this to the ballot box.

Principals: Tabulation tellers $TT_1, \ldots, TT_n$, broadcast bulletin board ABB, ballot boxes $VBB_1, \ldots, VBB_m$, supervisor Sup.
Public input: $X_{TT}$, contents of bulletin board ABB.
Private input to $TT_i$: Private key share $x_i$ of $X_{TT}$.
Output: Election tally for one block.
1. Each ballot box $VBB_i$ posts commitments on the list of all votes on the tabulation board ABB.
2. The supervisor signs the list of all received VBB commitments.
3. The tabulation tells $TT_i$ jointly execute 4–11.
4. Retrieve votes. Retrieve all votes from all endorsed ballot boxes $VBB_i$. Verify the commitments. Let $A \leftarrow$ list of votes.
5. Check proofs. Verify all VotePfs and ReencPfs in retrieved votes. Eliminate any votes with an invalid proof. Let $B$ be the list of remaining votes.
6. Duplicate elimination. Run the plaintext equivalence test $\text{PET}(S'_i, S'_j)$ for all pairs $(i, j)$, where $S'_x$ is the encrypted credential in vote $B_x$. Eliminate equivalent votes according to a revoting policy. Let $C$ be the list of remaining votes.
7. Mix votes. $D \leftarrow \text{MixNet}(C)$.
8. Mix credentials. Let $E$ be the list of all initially created encrypted credentials. Anonymize it: $F \leftarrow \text{MixNet}(E)$. 
9. **Invalid elimination.** Run the plaintext equivalence test PET($S_i, S'_j$) for all pairs $(i, j)$ where $S_i = F_i$, $S_j = D_j$. Eliminate votes from $D$ for which there is no equivalent credential found in $F$. Let $G$ be the list of remaining votes.

10. **Decrypt.** Let $H_i \leftarrow \text{DistDec}(G_i)$ for all $i$.

11. **Tally.** Compute the tally of $H$ according to an election method specified by the supervisor.

12. Finally, the supervisor endorses the tally (if ...).