UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Annabell Kuldmaa

On Secure Bulletin Boards for E-Voting

Master's Thesis (30 ECTS)

Supervisor: Helger Lipmaa

Tartu 2017

On Secure Bulletin Boards for E-Voting

Abstract: Vote collection together with storage of collected votes is the first phase of practically any electronic voting (e-voting) protocol. This functionality is provided by a *bulletin board* system. Many research papers in e-voting require the existence of a secure bulletin board, but there are only a few concrete systems. In the literature it is common to assume that bulletin board is a centralized trusted party, but in recent works the importance of a distributed fault-tolerant bulletin board has been raised.

In this thesis, we propose a formal model for analysis of security and functionality of a bulletin board system motivated by the security requirements Culnane and Schneider introduced in Computer Security Foundations Symposium 2014. We consider a secure bulletin board as a *robust public transaction ledger* presented by Garay et al. in Eurocrypt 2015 that additionally provides receipts for successful postings. More precisely, we introduce two properties: *(Confirmable) Persistence* and *Confirmable Liveness*.

We study a bulletin board system proposed by Culnane and Schneider in our model, and show that their protocol does not achieve Confirmable Liveness if there exist corrupted *item collection peers*, but achieves Confirmable Persistence for $< N/3$ corrupted item collection peers using only our trivial threshold signature scheme, otherwise the bound is $< N/4$. Motivated by the security analysis of Culnane-Schneider bulletin board system, we propose a fully secure bulletin board system and prove that it tolerates $< N/3$ corrupted item collection peers for Confirmable Persistence and $< N/2$ corrupted item collection peers for Confirmable Liveness.

This thesis is based on a submitted paper "A Cryptographic Approach to Bulletin Boards" with co-authors Aggelos Kiayas, Helger Lipmaa, Janno Siim and Thomas Zacharias.

Keywords: cryptography, e-voting, bulletin board, formal model

CERCS: P170 Computer science, numerical analysis, systems, control

E-valimiste jaoks mõeldud turvalistest teadetetahvlisüsteemidest

Lühikokkuvõte: Peaaegu iga elektroonilise hääletamise protokolli esimeseks etapiks on häälte kogumine ning nende talletamine. Seda teenust pakub *teadetetahvlisüsteem (bulletin board)*. Paljud teadusartiklid eeldavad turvalise teadetetahvlisüsteemi olemasolu, kuid konkreetseid süsteeme on välja pakutud vähe. Tihti eeldatakse, et teadetetahvlisüsteem on tsentraalne usaldatav osapool, kuid hiljutistes töödes on tähelepanu juhitud tõrkekindla hajustalletuse olulisusele.

Käesolevas töös pakume välja formaalse mudeli teadetetahvlisüsteemi funktsionaalsuse ning turvalisuse analüüsimiseks. Meie mudeli aluseks on Culnane ja Schneideri poolt konverentsil Computer Security Foundations Symposium 2014 väljapakutud teadetetahvlisüsteemi omadused. Me käsitleme turvalist teadetetahvlisüsteemi kui Garay ja teiste poolt konverentsil Eurocrypt 2015 tutvustatud *avalikku tehingute pearaamatut*, mis õnnestunud hääle talletamise korral väljastab kviitungi. Täpsemalt, me defineerime omadused *(tõendatav) püsivus* ning *tõendatav elusus*.

Me analüüsime Culnane ja Schneideri väljapakutud teadetetahvlisüsteemi turvalisust ning näitame, et nende protokolli korral ei ole elususe omadus täidetud, kui mõni *kogumisneel (item collection peer)* on ebaaus. Nende süsteem saavutab tõendatava püsivuse kasutades triviaalset lävisignatuuri juhul, kui ebaausaid kogumisneele on $< N/3$, vastasel korral on tõke $< N/4$. Culnane ja Schneideri teadetetahvlisüsteemist motiveeritult pakume välja uue süsteemi, mille korral on tagatud nii tõendatav püsivus kui ka tõendatav elusus, kui ebaausaid kogumisneele on vastavalt $< N/3$ ning $< N/2$. Lisaks on meie protokoll lihtne suhtluskeerukuselt.

Antud töö põhineb konverentsile esitatud artiklil „A Cryptographic Approach to Bulletin Boards"(„Krüptograafiline lähenemine teadetetahvlisüsteemidele"), mille kaasautoriteks on Aggelos Kiayas, Helger Lipmaa, Janno Siim ja Thomas Zacharias.

Võtmesõnad: krüptograafia, e-hääletamine, teadetetahvlisüsteem, formaalne mudel

CERCS: P170 Arvutiteadus, arvanalüüs, süsteemid, kontroll

Acknowledgments

In addition to my supervisor, I would like to thank Janno Siim and Thomas Zacharias for helpful discussions and inspiration. Janno's advise regarding security proofs, and Thomas' experience in e-voting and distributed systems were really valuable.

The author was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 653497 (project PANORAMIX).

Contents

1 Introduction **6**

2 Preliminaries **9**
2.1 Hash Functions 10
2.2 Signature Schemes 10
2.3 Threshold Signature Schemes 11

3 Security Framework for Bulletin Boards **14**
3.1 Entities and Protocols of a Bulletin Board System 14
3.2 Cryptographic Framework 15

4 Bulletin Board of Culnane-Schneider **20**
4.1 Description of the System 20
4.2 Security Analysis 22
4.2.1 Attacking Confirmable Persistence 22
4.2.2 Attacking Confirmable Liveness 23
4.3 Confirmable Persistence of Culnane-Schneider Bulletin Board System 25

5 A Fully Secure Bulletin Board System **28**
5.1 Description of the System 28
5.2 Confirmable Persistence 29
5.3 Confirmable Liveness 31

6 Related Work **36**
6.1 Byzantine Agreement Problem 36
6.2 Peters' Bulletin Board 36
6.3 Heather and Lundin's Bulletin Board 37
6.4 Krummenacher's Bulletin Board 38
6.5 Bulletin Board of STAR-Vote 38
6.6 Bulletin Board of D-DEMOS 39
6.7 E-Voting Service of Dini 40

7 Conclusion **41**

1 Introduction

Electronic voting (e-voting) is one crucial aspect of electronic governance. E-voting is expected to reduce the cost of elections and increase the voter turnout having the overall goal to verify the entire election procedure. Two types of e-voting can be identified: e-voting that uses voting machines located at polling stations and is physically supervised by electoral authorities; e-voting via the Internet, also called Internet voting (I-voting), where voting is done remotely using an electronic device connected to the Internet.

In high-level description we may divide the server-side of e-voting into three main stages. In the first stage, the task is to receive ballots and store them until the end of the election period. In the second stage, we need to anonymise the ballots and, finally, we must decrypt the encrypted ballots. These steps strongly depend on the approach chosen, e.g., in case of code-voting [Cha01] the first and the second steps cannot be distinguished (e.g., see [CZZ+16]).

The second phase can be implemented using mix-nets [Cha81]. Mix-net is a set of mix servers that guarantee anonymity by mixing the input in a verifiable way. Many implementations of mix-nets suitable for this scenario have been introduced, e.g., [Nef01], [Fur05], [TW10], [LZ12], [FL16], [FLZ16], and it is stated that to guarantee the anonymity of voters at least one of the mix servers must be uncorrupted.

The first phase can be implemented using a *bulletin board (BB)* system. Namely, it involves users who submit their ballots, *vote (item) collection (IC) peers* who receive and store the ballots, and a *web bulletin board (WBB)* that publishes the collected votes. The bulletin board should function as a ledger, i.e., once submitted and published, nothing can be erased.

Many research papers in e-voting require the existence of a secure BB, but there are only a few concrete systems (see [Pet05], [HL09] for early proposals). The authors of [CGS97] claim that BB can be viewed as a secure broadcast channel and, therefore, it can be implemented as a set of replicated servers implementing a Byzantine Agreement protocol. Although the latter claim is widely spread, the implementation of a BB is not straightforward. There are only a few approaches introduced which, in fact, are not purely Byzantine Agreement protocols. It is also common in the literature to assume that BB is a centralized trusted party, but in recent works ([CS14b], [CZZ+16]) the importance of a distributed fault-tolerant BB has been raised.

Results presented in this thesis are joint work with Aggelos Kiayas, Helger Lipmaa, Janno Siim and Thomas Zacharias. Furthermore, our research paper is submitted to a computer security conference.

The goal of this thesis (and our paper) is to study the security of BB systems for e-voting. Motivated by the requirements presented by Culnane and Schneider

in [CS14b], we introduce a formal framework consistent with the standards of distributed system research and cryptographic modelling to analyse the functionality and security of BB systems. We build a connection between blockchain and BB protocols, and formalize a secure BB system as a *robust public transaction ledger* introduced in [GKL15] that additionally provides receipts for successful postings. We define two properties named *Persistence* and *Confirmable Liveness*. The Persistence property intuitively means that in order to be published on the WBB, a threshold set of IC peers must approve the item, and once an item is published it cannot be removed from the WBB. In high-level, the Liveness property means that the protocol will successfully terminate and Confirmability captures the generation of receipts. Furthermore, we want Persistence to be Confirmable, i.e., the behaviour of corrupted WBB can be detected using a verification algorithm.

We study the security of Culnane-Schneider (C-S) BB system introduced in [CS14b] which uses a threshold signature scheme ([Des88], [Bol03]) to generate receipts. We show that the protocol does not achieve liveness in case we have corrupted IC peers, but it achieves Confirmable Persistence for threshold $< N/3$ of corrupted IC peers using the trivial threshold signature scheme (i.e., signature is a set of "normal" digital signatures). In case of non-trivial threshold signatures with distributed key generation, the threshold of corrupted IC peers is roughly $< N/4$.

Based on the analysis of C-S, we propose a fully secure BB system that achieves Confirmable Persistence and Confirmable Liveness for $< N/3$ and $< N/2$ dishonest IC peers, respectively. The enhanced protocol is very simple from the communication complexity aspect.

Contributions of the Author: In our paper, the author of this thesis researched thoroughly available literature on BB systems, including various e-voting protocols. The author compared available protocols and identified requirements for a secure BB system that could be adaptable for different e-voting systems.

The author was involved in developing a formal framework for secure BB systems. Namely, the author introduced the notation of Confirmable Persistence, i.e., the importance of having security against corrupted WBB in Persistence. Furthermore, the author observed that we cannot achieve both, Liveness and Soundness, thus, the Persistence considers only Stability and Unremovability.

The author also described the C-S BB system more formally and found the described weaknesses. More precisely, the author constructed the attacks and observed that the protocol does not achieve liveness in case of corrupted IC peers. Also, the author proved Confirmable Persistence and observed that it is achieved for $< N/3$ corrupted IC peers only using the trivial threshold signature scheme.

The author contributed to introducing a fully secure BB system. The author of this thesis observed that re-broadcasting the item in the Publishing protocol means

that we do not need to execute the Optimistic and Fallback protocols, it is sufficient that IC peers (threshold)sign their local BB records and send these signatures to the WBB. This made our BB system very simple from the communication complexity aspect. Furthermore, the author showed that using the trivial threshold signature scheme, the proposed BB system achieves Persistence and Confirmable Persistence tolerating $< N/3$ corrupted IC peers. The author of this thesis also observed that our fully secure BB system achieves Confirmable Liveness for $< N/2$ corrupted IC peers.

In comparison to the submitted paper, in this thesis we study C-S BB system more thoroughly: we present its detailed description and its formal security proof in our model. Furthermore, security proofs for our fully secure BB system are more detailed, and in this thesis, we give a good overview of other BB systems introduced in the literature. Furthermore, we also provide the reader with backround in distributed systems and cryptography.

Thesis Outline: We begin in Section 2 with introducing the notation and definitions used in the rest of the thesis. In Section 3 we present a cryptographic framework to analyse BB security. This is followed in Section 4 with describing the C-S BB system [CS14b], its security analysis and a security proof for Confirmable Persistence. In Section 5 we propose an enhanced BB system inspired by the C-S protocol and prove its security in our model. We conclude in Section 6 with an overview of related work on BB systems.

2 Preliminaries

In this section, we introduce the notation and terminology used in the rest of this thesis.

Consider a distributed system with *global clock* $\mathsf{Clock} \in \mathbb{N}$, each entity X has its *internal clock* $\mathsf{Clock}[X] \in \mathbb{N}$.

Definition 2.1 ([HT94])**.** *We say that a distributed system is synchronous if the following conditions hold:*

(i) For each step of a process there exist bounds for execution.

(ii) If an entity X sends a message to entity Y when $\mathsf{Clock} = T$, then entity Y receives it at some time $\mathsf{Clock} = T + \delta$ (eventual message delivery)*.*

(iii) For each entity X, it holds that $|\mathsf{Clock}[X] - \mathsf{Clock}| \leq \Delta$ (loose clock synchronization)*.*

Analogously, a distributed system is defined as asynchronous if there are no assumptions on process execution speeds, message transmission delays and clock drift rates.

A polynomial time Turing machine halts in polynomial time for any input. A Turing machine M is *Probabilistic Polynomial Time* (PPT) if it halts in polynomial time for any input and additionally has a randomness tape.

In security definitions we consider security games between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. We require that the advantage, i.e., the probability that any PPT adversary $\mathcal{A}$ wins is negligible in the security parameter.

Definition 2.2. *We say that a function ε is negligible, iff for any $c \in \mathbb{N}$ there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $\varepsilon(n) < n^{-c}$ holds.*

Let κ be the security parameter and denote a negligible function by $\mathbf{negl}(\cdot)$.

An adversary $\mathcal{A}$ is defined as *static* if it must select the set of players to corrupt before the execution of the computation and *adaptive* if the adversary chooses who to corrupt during the computation. We say that an adversary $\mathcal{A}$ is *covert* [AL10] if it may deviate arbitrarily from the protocol specification in an attempt to cheat, but does not want to get "caught". Furthermore, we consider an adversary $\mathcal{A}$ as *stateful* if it maintains a state across queries. In the following of this thesis, we consider PPT adversaries.

Bitcoin [Nak08] is a decentralized payment system based on a public transaction ledger in distributed manner. Anynomous players called *miners* execute a protocol that maintains and extends the *blockchain*. Garay et al. introduced in [GKL15] the notation of a *public transaction ledger* that captures the essence

of Bitcoin's operation as cryptocurrency. Namely, it guarantees two properties of committed transactions: *Persistence* and *Liveness*. Persistence means that once a transaction goes more than k blocks "deep" into the blockchain of one honest player, then with overwhelming probability it will be included in blockchain of every honest player. Furthermore, the ledger will assign that traction a permanent position. In other words, all honest players will agree on all transactions and their order.

Liveness property claims that all transactions originating from honest account holders will eventually end up at a depth more than k blocks in blockchain of an honest player, meaning that (honest) transactions will eventually be inserted into the blockchain. In Section 3 we expand this notation for our formal framework for secure BB systems.

In [Lam77] *liveness* property is informally defined as a requirement that something good will eventually happen. Therefore, we may say that a protocol achieves liveness if it eventually terminates. In this work, we consider Confirmable Liveness described in Section 3.2.

2.1 Hash Functions

Hash functions are functions that compress an input of arbitrary length to a result with fixed length. If hash functions satisfy additional requirements, they are a very powerful tool in the design of techniques to protect the authenticity of information. A formal definition for a collision-resistant hash function family was given in [Dam88].

Definition 2.3. *We say that $H : \{0,1\}^* \times K \to \{0,1\}^n$ is a collision-resistant hash function family if for random $k \leftarrow_u K$, it is hard to find $x_0, x_1 \in \{0,1\}^*$ such that $H_k(x_0) = H_k(x_1)$, i.e., for any PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathbf{negl}(\cdot)$ *such that for all $\kappa \in \mathbb{N}$,*

$$\Pr\left[k \leftarrow_u K, (x_0, x_1) \leftarrow \mathcal{A}(1^\kappa, H, k) : H_k(x_0) = H_k(x_1)\right] \leq \mathbf{negl}(\kappa).$$

2.2 Signature Schemes

(Threshold) signature schemes can take a set of global parameters (e.g., the description of the underlying group) as an additional input. For simplicity, we omit this in the following.

Definition 2.4. *A digital signature scheme consists of three efficient algorithms, denoted by $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$.*

$\mathsf{KGen}(1^\kappa)$**:** *A generation algorithm* KGen *generates a keypair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\kappa)$ *for a secret signing key* sk *and a public verification key* pk*.*

$\mathsf{Sig}_{\mathsf{sk}}(m)$**:** *A signing algorithm* Sig *inputs a message* m *and a secret signing key* sk*, and returns a signature* $\sigma \leftarrow \mathsf{Sig}_{\mathsf{sk}}(m)$ *on the message* m*.*

$\mathsf{Vf}_{\mathsf{pk}}(m, \sigma)$**:** *A verification algorithm* Vf *takes a public verification key* pk *and a message* m *with corresponding signature* σ *as input, and returns* $b \leftarrow \mathsf{Vf}_{\mathsf{pk}}(m, \sigma)$ *where* $b \in \{0, 1\}$*.*

For each $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{KGen}(1^\kappa)$ *and a valid message* m*, it holds that*

$$\mathsf{Vf}_{\mathsf{pk}}(m, \mathsf{Sig}_{\mathsf{sk}}(m)) = 1.$$

We require the signature scheme to be existentially unforgeable against chosen message attacks. The *existential unforgeability against chosen message attack* (EUF-CMA, [GMR88]) security property is defined via the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Definition 2.5. *We say that a signature scheme* Σ *is EUF-CMA-secure if*

$$\mathsf{Adv}^{\text{forge}}(\mathcal{A}) = \Pr\left[\mathcal{G}_\Sigma^{\mathcal{A}}(1^\kappa) = 1\right] \leq \mathbf{negl}(\kappa),$$

for any PPT adversary $\mathcal{A}$ *where*

$\mathcal{G}_\Sigma^{\mathcal{A}}(1^\kappa)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\kappa)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sig}_{\mathsf{sk}}(\cdot)}(\mathsf{pk})$
if m^**has been queried from* $\mathsf{Sig}_{\mathsf{sk}}(\cdot)$ return $\perp$
return $\mathsf{Vf}_{\mathsf{pk}}(m^*, \sigma^*)$.

Informally, in the Definition above, we give the adversary $\mathcal{A}$ access to the signing oracle $\mathsf{Sig}_{\mathsf{sk}}(\cdot)$, and say that it wins if it manages to forge a signature on some message m^*, i.e., $\mathsf{Vf}_{\mathsf{pk}}(m^*, \sigma^*) = 1$, without querying it from the signing oracle.

2.3 Threshold Signature Schemes

Before we define threshold signature schemes (TSS), we informally introduce the concept of threshold secret sharing [Sha79]. Secret sharing is a well-known cryptographic method for protecting the secret s by distributing it amongst a set of N parties.

Let t, N be two positive integers such that $t < N$. The set of values $(s_1, \ldots, s_N)$ is said to be a (t, N)*-threshold secret sharing* of the value of s, if no t values from this set reveal any information about s while there exists an efficient algorithm that takes as an input $t + 1$ values from this set and outputs s. We denote it $(s_1, \ldots, s_N) \overset{(t,N)}{\rightarrow} s$.

Definition 2.6 ([Des88]). *Let $t_s < N$ be two positive integers. Assume the existence of a (t_s, N)-threshold secret sharing scheme.*

A (non-interactive) threshold signature scheme TSS consists of five efficient algorithms, denoted by $\mathsf{TSS} = (\mathsf{DistKeygen}, \mathsf{ShareSig}, \mathsf{ShareVerify}, \mathsf{TssVf}, \mathsf{Combine})$.

$\mathsf{DistKeygen}(1^\kappa, t_s, N)$**:** *A randomized distributed interactive threshold key generation protocol that is run by the parties $P_1, \ldots, P_N$. The public output is* pk *and a tuple of public verification keys* $(\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$. *The private output of each P_i is a value* ssk_i *such that* $(\mathsf{ssk}_1, \ldots, \mathsf{ssk}_N) \overset{(t_s,N)}{\rightarrow} \mathsf{ssk}$, *where* ssk *is the secret key corresponding to* pk.

$\mathsf{ShareSig}_{\mathsf{ssk}_i}(m)$**:** *A possibly randomized algorithm that takes as an input a private key share* ssk_i *and a message m outputting a signature share σ_i on the message m.*

$\mathsf{ShareVerify}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_N, m, (i, \sigma_i))$: *Outputs $b \in \{0, 1\}$, depending on whether σ_i is a valid i-th signature share on m.*

$\mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_N, m, (i, \sigma_i)_{i \in S})$**:** *Given that $S \subseteq I$ is a subset of $t_s + 1$ elements and σ_i is the ith signature share on m, outputs a full signature* $\sigma = \mathsf{TSign}_{\mathsf{ssk}}(m)$ *on m. (Or outputs $\perp$, when some signature shares are ill-formed.)*

$\mathsf{TssVf}_{\mathsf{pk}}(m, \sigma)$**:** *A deterministic algorithm that outputs $b \in \{0, 1\}$, depending on whether σ is a valid signature on m.*

For $(\mathsf{ssk}, \mathsf{pk}, \mathsf{ssk}_1, \ldots, \mathsf{ssk}_N, \mathsf{pk}_1, \ldots, \mathsf{pk}_N)$ *output by* $\mathsf{DistKeygen}(1^\kappa, t_s, N)$, *if $S \subseteq I$ such that $|S| = t_s + 1$,* $\sigma_i = \mathsf{ShareSig}_{\mathsf{ssk}_i}(m)$, *and* $\sigma = \mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_N, m, (i, \sigma_i)_{i \in S})$, *and a valid message m, it holds that*

$$\mathsf{ShareVerify}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_N, m, (i, \sigma_i)) = 1, \forall\, i \in S,$$
$$\mathsf{TssVf}_{\mathsf{pk}}(m, \sigma) = 1.$$

The *existential (t_s, N)-unforgeability against chosen message attacks*((t_s, N)-EUF-CMA) security property is defined via the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Definition 2.7. *We say that a threshold signature scheme* TSS *is (t_s, N)- EUF-CMA-secure against static corruption if*

$$\mathsf{Adv}^{\text{forge}}(\mathcal{A}) = \Pr\left[\mathcal{G}^{\mathcal{A}}_{\mathsf{TSS}}(1^\kappa) = 1\right] \leq \mathbf{negl}(\kappa),$$

for any stateful PPT adversary $\mathcal{A}$ where

$$
\begin{array}{l}
\mathcal{G}^{\mathcal{A}}_{\mathsf{TSS}}(1^{\kappa}) \\
\left[\begin{array}{l}
\mathcal{C} \subset \{1,\ldots,N\} \leftarrow \mathcal{A} \\
(\mathsf{ssk}, \mathsf{pk}, \mathsf{ssk}_1, \ldots, \mathsf{ssk}_N, \mathsf{pk}_1, \ldots, \mathsf{pk}_N) \leftarrow \mathsf{DistKeygen}(1^{\kappa}, t_s, N) \\
(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{ShareSig}_{\mathsf{ssk}_j}(\cdot)}(\mathsf{pk}, \{\mathsf{pk}_j\}),\ j \in \{1,\ldots,N\} \\
\mathcal{V} = \mathcal{C} \cup \mathcal{S}, \mathcal{S} = \{i : \mathcal{A} \textit{ made a signing query on } (i, m^*)\} \subset \{1,\ldots,N\} \\
\mathsf{if}\ |\mathcal{V}| \geq t_s + 1\ \mathsf{return}\ \perp \\
\mathsf{return}\ \mathsf{TSSVf}_{\mathsf{pk}}(m^*, \sigma^*).
\end{array}\right.
\end{array}
$$

Informally, in the Definition above, we let the adversary $\mathcal{A}$ choose the set $\mathcal{C}$ of parties to corrupt and give it access to the signing oracle $\mathsf{ShareSig}_{\mathsf{ssk}_j}(\cdot)$, $j \in \{1,\ldots,N\}$. We say that $\mathcal{A}$ wins if it manages to forge a signature on some message m^*, i.e., $\mathsf{TssVf}_{\mathsf{pk}}(m^*, \sigma^*) = 1$, with seeing no more more than $t_s = |C \cup S|$ signature shares where $\mathcal{S}$ is a set of parties i such that $\mathcal{A}$ made a signing query on (i, m^*).

Proposed (t_s, N)-EUF-CMA-secure threshold signature schemes having distributed key generation, e.g., [Bol03], [LJY14], require $t_s < N/2$.

We note that some threshold signature schemes [Sho00, AMN01] consider the case of (k, t_s, N)-threshold signature schemes, where t_s is the number of corrupted parties the scheme can tolerate, and to actually honestly give a signature, the participation of $k \geq t_s + 1$ parties is required. Such schemes allow one to have $k > N/2$, but they are based on a trusted dealer. Having a trusted dealer should be avoided in e-voting scenario.

In this work, we also use the following *trivial* (t_s, N)-*TSS*, where $t_s < N$ can be any integer: given an EUF-CMA-secure signature scheme Σ, each party P_i has independently generated public and secret keys. The signature share of P_i on m is equal to its signature on m. Signature shares of different P_i on m are combined by concatenating them. The TSS verification on $(\sigma_1, \ldots, \sigma_t)$ succeeds only if each individual signature verification succeeds and $t > t_s$. Clearly, this trivial TSS is (t_s, N)-EUF-CMA-secure. Observe that in case of trivial TSS we do not have that ssk is secret shared as stated in Definition 2.7. We have that $\mathsf{ssk} = \{\mathsf{ssk}_i\}, i \in S \subseteq \{1,\ldots,N\}$ such that $|S| \geq t_s + 1$. However, we still consider it as a trivial TSS.

3 Security Framework for Bulletin Boards

In this section, we present a cryptographic framework for BB systems that was introduced in our paper. We begin with describing the parties and the protocols involved. This is followed by introducing our framework based on security requirements from [CS14b] and modelling approach presented in [CZZ+16] and [GKL15].

3.1 Entities and Protocols of a Bulletin Board System

A BB system consists of a number of parties:

- The *setup authority* denoted by SA is responsible for generating the setup information and initializes all other parties with their private inputs.
- The *users* that submit items. In case of e-voting, items can be voters' ballots, elections result or audit information. Let n denote the number of users.
- The *item collection (IC) peers* denoted by $\mathsf{P}_1, \ldots, \mathsf{P}_{N_c}$ that interact with users to post items and with the WBB to publish items.
- The *web bulletin board (WBB) peers* denoted by $\mathsf{WBB}_1, \ldots, \mathsf{WBB}_{N_w}$ are responsible for publishing posted items. In general, we consider the WBB as a distributed sub-system, although in the literature, e.g., [CS14b], the WBB sometimes contains only one publicly accessible peer.

During the setup SA specifies a *posting policy* denoted by

$$\mathcal{P} = (\mathsf{Accept}(\cdot,\cdot), \mathsf{S}(\cdot)),$$

where $\mathsf{Accept}(\cdot,\cdot)$ is a binary relation defined over pairs of user IDs and items, and $\mathsf{S}(\cdot)$ is a *selection function* over sets of items. More precisely, for user U to post an item x, $(\mathsf{U}, x) \in \mathsf{Accept}$ must hold. Function S is responsible for resolving conflicts between clashing items. It takes as input a set X_{U} of all items published by user U and outputs a valid item x, e.g., in Estonian e-voting [HW14] if a voter U posted m items $X_U = \{x_1, \ldots, x_m\}$, then only the last one must count. We may write $\mathsf{S}(X_{\mathsf{U}}) = \{x_m\}$.

The setup authority initializes all entities with description of $\mathcal{P}$. Next, all parties engage in the setup interaction and, thus, every party has a private input, e.g., signing key of a signature scheme over standard public key infrastructure, or an authentication password, and some public parameters denoted by params.

We capture the functionality of BB using the following protocols:

- The *Posting* protocol is started by a user U to post item x using private input s_{U}. The user U and IC peers interact and successful posting of x results in user U obtaining a receipt $\mathrm{rec}[x]$.

- The *Publishing* protocol in which IC peers upload their local records of posted items to the WBB.

There are two verification algorithms associated with the protocols: $\mathsf{VerifyRec}$ takes $(\mathsf{rec}[x], x, s_{\mathsf{U}}, \mathsf{params})$ as input and is run by users to verify successful postings; $\mathsf{VerifyPub}$ takes the published data and params as public input and is used to check the WBB consistency.

Following [CZZ+16], we assume the existence of a *global clock* denoted by $\mathsf{Clock} \in \mathbb{N}$ and that each system entity X has an *internal clock* denoted by $\mathsf{Clock}[X] \in \mathbb{N}$. We define the following events on the clocks:

- The event $\mathsf{Init}(X) : \mathsf{Clock}[X] \leftarrow \mathsf{Clock}$, that initializes an entity X by synchronizing its internal clock $\mathsf{Clock}[X]$ with the global clock Clock.
- The event $\mathsf{Inc}(\mathsf{Clock}[X]) : \mathsf{Clock}[X] \leftarrow \mathsf{Clock}[X] + 1$, that causes the internal clock of entity X to advance by one time unit.

Let $\mathbf{E} := \{\mathsf{SA}\} \cup \{\mathsf{U}_\ell\}_{\ell \in [n]} \cup \{\mathsf{P}_i\}_{i \in [N_c]} \cup \{\mathsf{WBB}_j\}_{j \in [N_w]}$ be the set of all entities of a BB system. We denote by t_c, t_w the number of peers that the adversary may statically corrupt and by N_c, N_w the total peers of the IC and WBB subsystem, respectively. A corrupted peer may deviate arbitrarily from the protocol, or even go offline again.We denote the local record of an IC peer P_i at internal time $\mathsf{Clock}[\mathsf{P}_i] = T$ as the set of items $L_{\mathsf{post},i,T}$. The set of items published by WBB peer WBB_j at internal time $\mathsf{Clock}[\mathsf{WBB}_j] = T$ is denoted by $L_{\mathsf{pub},j,T}$.

3.2 Cryptographic Framework

The authors of [CS14b] claim that their BB must satisfy the following properties which we refer to as *Stability, Confirmability, Soundness* and *Unremovability*:

- (bb.1) Only items that have been posted can appear on the WBB (Stability).
- (bb.2) Any item with a valid receipt must appear on the WBB (Confirmability).
- (bb.3) No clashing items must both appear on the WBB (Soundness).
- (bb.4) Once published, no items can be removed from the WBB (Unremovability).

The aforementioned properties are not formal enough to study cryptographic security of the system. In the following, we formalise these properties. Namely, study these properties under a formal framework inspired by the approach of distributed e-voting security of D-DEMOS [CZZ+16] with the notation of a *robust*

public transaction ledger (RPTL) introduced in [GKL15]. The idea is to model a secure BB as an RPTL that provides receipts for successful postings. The security properties of an RPTL are captured using *Persistence* and θ-*Liveness*.

We introduce the concept of *(Confirmable) Persistence*. Persistence defined in [GKL15] expresses the Unremovability property. In order to capture Stability, we additionally require that an item x posted by user U cannot appear on the WBB if it is was not added to local BB for at least $N_c - 2t_c$ honest IC peers during the Posting protocol. We extend Persistence by considering the case that the WBB sub-system can be also corrupted by a covert adversary. This is captured by *Confirmable Persistence* where we require that any malicious behaviour of the WBB is detected via the $\mathsf{VerifyPub}$ algorithm.

Confirmable θ-*Liveness* means that any honest user that submits an item x will obtain a receipt within time θ and x will be published on the WBB. We observe that Soundness and Confirmable Liveness cannot be satisfied concurrently if honest users may post clashing items (attack in Section 4.2.1), e.g., in Estonian e-voting [HW14] as a coercion countermeasure voters are allowed to vote multiple times during the election period. This issue can be resolved via selection function S. More precisely, we do not require the Soundness property to hold and allow users to post clashing items, but we use verifiable S to remove all conflicting votes. This allows us to achieve Confirmable Liveness.

In the threat model for Persistence, we assume asynchronous communication allowing adversary to also drop messages between honest parties. The Persistence property is formally defined via a security game $\mathcal{G}^{\mathcal{A},t_c,t_w}_{\mathsf{Prst}}\big(1^\kappa, \mathbf{E}\big)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The game is described in Figure 1. Informally it means that the adversary $\mathcal{A}$ wins there exists a published item x which is not in local BB records for at least $N_c - 2t_c$ honest IC peers, or at some time T an item x was published on the WBB, but at some $T' > T$ it has been removed.

In Confirmable Persistence we allow the WBB to be malicious and deviate from the Publishing protocol. Confirmable Persistence is formally defined via a security game $\mathcal{G}^{\mathcal{A},t_c}_{\mathsf{C.Prst}}\big(1^\kappa, \mathbf{E}\big)$ that follows the steps from $\mathcal{G}^{\mathcal{A},t_c,t_w}_{\mathsf{Prst}}\big(1^\kappa, \mathbf{E}\big)$ for a special case where $t_w = N_w$, except for the following:

- The inconsistent $\mathsf{WBB_j}$ referred in either (P.1) or (P.2) may be any corrupted WBB peer.
- For all $j \in \{1, \ldots, N_w\}$ and every moment T it must hold that $\mathsf{VerifyPub}\ (L_{\mathsf{pub},j,T}, \mathsf{params}) = \mathsf{accept}$.

Informally it means that verification algorithm $\mathsf{VerifyPub}$ must always accept.

Given the security games, Persistence and Confirmable Persistence are defined as follows.

Threat model for Persistence.

(P.I). The adversary $\mathcal{A}$ statically corrupts up to t_c, t_w out-of the N_c, N_w total peers of the IC and WBB sub-systems, respectively. Then, $\mathcal{A}$ provides the challenger $\mathcal{C}$ with the set $L_{\mathsf{corr}} \subset \mathbf{E}$ of corrupted parties. Throughout the game, $\mathcal{C}$ plays the role of honest entities that include SA.

(P.II). When an honest entity X wants to transmit a message $\mathbf{M}$ to an honest entity Y, then it sends $(X, \mathbf{M}, Y)$ to $\mathcal{A}$, which may not forward $\mathbf{M}$ to Y.

(P.III). $\mathcal{A}$ may write on the incoming network tape of any honest entity.

(P.IV). $\mathcal{A}$ may invoke the event $\mathsf{Inc}(\mathsf{Clock}[X])$ arbitrarily.

Protocol execution under the presence of $\mathcal{A}$.

- The challenger $\mathcal{C}$ initiates the setup phase playing the role of SA and determines the posting policy $\mathcal{P}$. It initializes every system entity $X \in \mathbf{E}$ by running the event $\mathsf{Init}(X)$.
- Upon initialization, $\mathcal{C}$ and $\mathcal{A}$ engage in the setup phase, and the Posting and Publishing protocols, where $\mathcal{C}$ acts on behalf of all honest entities.
- For each user U_ℓ, $\ell \in \{1, \ldots, n\}$, $\mathcal{A}$ may choose either of the two options:
  (a). Corrupt, and thus fully control U_ℓ.
  (b). Do not corrupt U_ℓ. In this case, $\mathcal{A}$ may initialize U_ℓ by running $\mathsf{Init}(\mathsf{Clock}[\mathsf{U}_\ell])$ only once.
- The adversary $\mathcal{A}$ may provide $\mathcal{C}$ with a message $(\mathsf{post}, \mathsf{U}_\ell, x)$ for some honest user U_ℓ and an item x of its choice. Upon receiving $(\mathsf{post}, \mathsf{U}_\ell, x)$, $\mathcal{C}$ engages in the Posting protocol on behalf of U_ℓ. If the interaction is completed successfully, $\mathcal{C}$ obtains a receipt $\mathsf{rec}[x]$ for x.

Game winning conditions.

The game outputs 1 iff there is a WBB peer $WBB_j \notin L_{\mathsf{corr}}$ such that at least one of the following holds:

(P.1). There is an item x, $t_c + 1$ IC peers $\{\mathsf{P}_{i_k}\}_{k \in [t_c+1]}$ and moments $\{T_{i_k}\}_{k \in [t_c+1]}$ s.t. for every $T' \geq \min_{k \in [t_c+1]} \{T_{i_k}\}$: (i) $\{\mathsf{P}_{i_1}, \ldots, \mathsf{P}_{i_{t_c+1}}\} \cap L_{\mathsf{corr}} = \emptyset$, (ii) for every $k \in [t_c + 1] : x \notin L_{\mathsf{post},i_k,T_{i_k}}$ and (iii) $x \in L_{\mathsf{pub},j,T'}$ (*Stability attack*).

(P.2). There is an item x, and two moments T, T' s.t. (i) $T' > T$, (ii) $x \in L_{\mathsf{pub},j,T}$ and (iii) $x \notin L_{\mathsf{pub},j,T'}$(*Unremovability attack*).

Figure 1: The BB Persistence security game $\mathcal{G}^{\mathcal{A},t_c,t_w}_{\mathsf{Prst}}(1^\kappa, \mathbf{E})$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Threat model for Confirmable Liveness.

(L.I). As (P.I) in Figure 1.

(L.II). When an honest entity X wants to transmit a message $\mathbf{M}$ to an honest entity Y, then it just sends $(X, \mathbf{M}, Y)$ to $\mathcal{A}$. If the honest entity X sends $(X, \mathbf{M}, Y)$ to $\mathcal{A}$, when the global time is $\mathsf{Clock} = T$, then $\mathcal{A}$ must write M on the incoming network tape of Y by the time that $\mathsf{Clock} = T + \delta$ (*eventual message delivery*).

(L.III). As (P.III) in Figure 1.

(L.IV). $\mathcal{A}$ may invoke the event $\mathsf{Inc}(\mathsf{Clock}[X])$ under the restriction that for any entity X, $|\mathsf{Clock}[X] - \mathsf{Clock}| \leq \Delta$ (*loose clock synchronization*).

Protocol execution under the presence of $\mathcal{A}$.

- The challenger initiates the Setup phase playing the role of SA and determines the post access policy $\mathcal{P} = \big(\mathsf{Accept}(\cdot,\cdot), \mathsf{S}(\cdot)\big)$. Then, it initializes every system entity $X \in \mathbf{E}$ by running the event $\mathsf{Init}(X)$.
- Upon initialization, $\mathcal{A}$ and $\mathcal{C}$ engage in the Setup phase, the Posting, and Publishing protocols as in Figure 1.

Game winning conditions.

The game outputs 1 iff there is an honest user U and a moment T s.t. the following conditions hold:

(L.1). $\mathcal{A}$ provided $\mathcal{C}$ with the message $(\mathsf{post}, \mathsf{U}, x)$ at global time $\mathsf{Clock} = T$.

(L.2). No honest IC peer engages in the Publishing protocol during global time $\mathsf{Clock} \in [T, T + \theta]$.

(L.3). Either of the following is true:

(a). By internal time $\mathsf{Clock}[\mathsf{U}] \leq T + \theta$, $\mathcal{C}$ did not obtain a value z s.t. $\mathsf{VerifyRec}(z, x, \mathsf{cr}_{\mathsf{U}}, \mathsf{params}) = \mathsf{accept}$, or

(b). There is a $\mathsf{WBB}_j \notin L_{\mathsf{corr}}$ s.t. for any moment T_j, there is a moment $T'_j \geq T_j$ s.t. $x \notin L_{\mathsf{pub},j,T'_j}$.

Figure 2: The Confirmable θ-Liveness security game $\mathcal{G}^{\mathcal{A},\delta,\Delta,t_c,t_w}_{\theta-\mathsf{C.Live}}(1^\kappa, \mathbf{E})$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Definition 3.1 ((Confirmable) Persistence)**.** *Let κ be the security parameter and* **BB** *be a BB system with N_c IC peers and N_w WBB peers. We say that* **BB** achieves Persistence *for fault tolerance thresholds (t_c, t_w), if for every PPT adver-*

sary $\mathcal{A}$ it holds that

$$\Pr\left[\mathcal{G}^{\mathcal{A},t_c,t_w}_{\mathsf{Prst}}(1^\kappa, \mathbf{E}) = 1\right] \leq \mathbf{negl}(\kappa).$$

Furthermore, we say that **BB** achieves Confirmable Persistence *for fault tolerance threshold t_c, if for every PPT adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\mathcal{G}^{\mathcal{A},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E}) = 1\right] \leq \mathbf{negl}(\kappa).$$

In the threat model for Confirmable Liveness we consider synchronous communication with eventual message delivery and bounded synchronization loss with respect to the global clock, specified by upper bounds δ and Δ. Confirmable θ-Liveness is formally defined via a security game $\mathcal{G}^{\mathcal{A},\delta,\Delta,t_s,t_w}_{\Theta-\mathsf{C.Live}}\left(1^\kappa, \mathbf{E}\right)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ in Figure 2.

Intuitively it means that the adversary $\mathcal{A}$ wins if an honest user does not obtain a valid receipt within some θ, or an honestly posted item x is not published on the WBB.

Definition 3.2 (Confirmable θ-Liveness)**.** *Let κ be the security parameter, $\theta, \delta, \Delta \in \mathbb{N}$ and let* **BB** *be a BB system with with N_c BB peers and N_w WBB peers. We say that* **BB** achieves Confirmable θ-Liveness *for fault tolerance thresholds (t_c, t_w), delay message bound δ and synchronization loss bound Δ, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\Pr\left[\mathcal{G}^{\mathcal{A},\delta,\Delta,t_c,t_w}_{\theta-\mathsf{C.Live}}(1^\kappa, \mathbf{E}) = 1\right] \leq \mathbf{negl}(\kappa).$$

4 Bulletin Board of Culnane-Schneider

In this section, we describe the BB of vVote voting system ([CRST15, BCS16]) introduced by Culnane and Schneider in [CS14b]. In the following, we refer to it also as C-S BB system. The vVote voting system was used in the state election in the Australian state of Victoria in November 2014. We study the security of C-S BB as it has already been used in real-world elections and the vVote system itself has been published in IEEE Security and Privacy 2016 which is one of the best journals in the field. In Section 6 we describe other BB systems introduced in the literature. We have included related work in the end of this thesis, so that the concept would be more understandable and weaknesses clearer to the reader.

We begin with detailed overview of the C-S BB system followed by its security analysis.

4.1 Description of the System

The C-S BB uses a signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$ and a (t_s, N_c)-threshold signature scheme $\mathsf{TSS} = (\mathsf{DistKeygen}, \mathsf{ShareSig}, \mathsf{ShareVerify}, \mathsf{TssVf}, \mathsf{Combine})$. Denote by sk_i the individual signature key of peer P_i. Let ssk be the threshold signature key, and let ssk_i be its ith share. Denote by Clash some predefined clash relation and by H some hash function.

The C-S BB system considers users, IC peers denoted by $\mathsf{P}_i, i \in \{1, \ldots, N_c\}$, and a single WBB. We note that in [CS14b] IC peers are called local BB peers, for clarity to minimise confusion between BB peers and the WBB, we refer to IC peers and the WBB. Clearly, the system is initialized by SA but Culnane and Schneider do not address this in their paper. The C-S BB system runs in periods and consists of the Posting protocol and the Publishing protocol. The Publishing protocol consists of two protocols, the *Optimistic* protocol and the *Fallback* protocol. Each period p is a time interval between $T_{\mathsf{begin},p}$ and $T_{\mathsf{end},p}$, i.e., $p := [T_{\mathsf{begin},p}, T_{\mathsf{end},p}]$. For each period p an IC peer P_i has a local record $B_{i,p}$ of received items x and a local database $D_{i,p}$ of signatures on items x from other IC peers. In the beginning of each period p, each peer P_i has $B_{i,p}, D_{i,p} \leftarrow \emptyset$, i.e., both databases are initialized empty.

While the following protocols directly follow C-S BB, we include more details so it would be possible to analyse their security. C-S do not describe any Accept nor Clash relations. For clarity, we have added the credential cr_U for each user U and require that for user U to post an item x it holds that $\mathsf{Accept}(\mathsf{U}, x) = 1$. Furthermore, C-S simply claim that each P_i checks that there is no clash between x and previous posts, in the following we require that $\mathsf{Clash}(x, x') = 0$ for all previously posted x'. The following protocols use hash function H, but we note that hash functions were used in the full version [CS14a], but not in the conference

version of C-S BB [CS14b].

Assume that we have a period p. The protocol for user U to post an item x is as follows.

Posting Protocol:

$\mathsf{U} \to \mathsf{P}_i$**:** User U broadcasts item x with credential cr_U to all peers P_i, $i \in [N_c]$.

$\mathsf{P}_i \to \mathsf{P}_j$**:** Upon receiving x from U, each P_i checks that (i) cr_U is valid for U (i.e $\mathsf{Accept}(\mathsf{U}, x) = 1$) and (ii) x does not clash with any previously posted item x' during p or previous periods (i.e., $\mathsf{Clash}(x, x') = 0$). If both checks are successful, then it broadcasts $(i, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_i}(m))$ to all other IC peers.

P_i**:** P_i waits for messages $(j, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_j}(m))$ from peers P_j, $j \neq i$, and appends them to the dataset of received signed items $D_{i,p}$.
– Upon receiving $N_c - t_c$ valid signatures on $(p, x, \mathsf{cr}_\mathsf{U})$ (including its own), P_i adds (p, x) to its local BB record $B_{i,p}$ for period p.

$\mathsf{P}_i \to \mathsf{U}$**:** Upon adding (p, x) to $B_{i,p}$, P_i sends its TSS share $(i, m = (p, x), \mathsf{ShareSig}_{\mathsf{ssk}_i}(m))$ to U.

U**:** The user U waits for signatures $(j, m = (p, x), \mathsf{ShareSig}_{\mathsf{ssk}_j}(m))$ from $N_c - t_c \geq t_s + 1$ peers P_j, $j \neq i$. When this happens, let S_U be a set of $N_c - t_c$ peers from which U can combine the share. U obtains the receipt for x,
$\mathsf{rec}[x] := \mathsf{TSign}_{\mathsf{ssk}}((p, x)) \leftarrow \mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_{N_c}, (p, x), (j, \sigma_j)_{j \in S_\mathsf{U}})$.

Local BB records are published at the end of each period p. The Fallback protocol is executed only if the threshold of peers $(N_c - t_c)$ do not agree on local BB records in the end of the Optimistic protocol. If all peers are honest, the Fallback protocol is never executed.

Optimistic Protocol:

$\mathsf{P}_i \to \mathsf{P}_j$**:** Each P_i broadcasts $\big(i, m = (p, H(B_{i,p})), \mathsf{Sig}_{\mathsf{sk}_i}(m)\big)\big)$ to all other IC peers.
– Each P_i waits until it receives $N_c - t_c$ valid signatures $\big(j, m = (p, H(B_{j,p})), \rho_j := \mathsf{Sig}_{\mathsf{sk}_j,}(m)\big)$ from different IC peers (including its own).

– If $\sharp\left\{j \in I : H(B_{j,p}) = H(B_{i,p}) \wedge \mathsf{Vf}_{\mathsf{pk}_j}(\rho_j) = 1\right\} < N_c - t_c$, then P_i broadcasts a message $(i, m = (p, D_{i,p}), \mathsf{Sig}_{\mathsf{sk}_i}(m))$ to all IC peers, indicating that it engages in the Fallback protocol (see below).

$\mathsf{P}_i \to \mathsf{WBB}$: Each P_i sends $(i, m = (p, B_{i,p}), \mathsf{ShareSig}_{\mathsf{ssk}_i}(m))$ to WBB.

– WBB waits for messages $(j, m = (p, B_{j,p}), \sigma_j = \mathsf{ShareSig}_{\mathsf{ssk}_j}(m))$ from at least $N_c - t_c \geq t_s + 1$ peers P_j.

– Otherwise, let S be a set of those $\geq N_c - t_c$ peers that agree on the contents of the BB and let B_p be the board of agreed-on contents. Obtain $\mathsf{TSign}_{\mathsf{ssk}}((p, B_p)) \leftarrow \mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_{N_c}, (p, B_p), (j, \sigma_j)_{j \in S})$.

– Publish $\mathsf{WBBreceipt}[p, B_p] := (m = (p, B_p), \mathsf{TSign}_{\mathsf{ssk}}(m))$.

Fallback Protocol:

$\mathsf{P}_i \to \mathsf{P}_j$: Each P_i broadcasts $(i, m = (p, D_{i,p}), \mathsf{Sig}_{\mathsf{sk}_i,}(m))$ to all other IC peers.

– Each P_j, $j \neq i$, updates its database with the new data, i.e., with the signatures it is missing, and then updates its board. More precisely, if $\mathsf{Vf}_{\mathsf{pk}_i}\big(m = (p, D_{i,p}), \mathsf{Sig}_{\mathsf{sk}_i,}(m))\big) = 1$, then P_j will put all new signatures in $D_{i,p}$ to its database $D_{j,p}$ (i.e. P_j sets $D_{j,p} \leftarrow D_{j,p} \cup D_{i,p}$). For any x: if P_j has $N_c - t_c$ valid signatures on (p, x, cr_U) then he adds (p, x) to $B_{j,p}$.

– P_j broadcasts a message $\big(j, m = (p, H(B_{j,p})), \mathsf{Sig}_{\mathsf{sk}_j}(m)\big)$ indicating that it re-engages in Step 1 of the Optimistic protocol (see above) for the updated record $B_{j,p}$.

4.2 Security Analysis

The authors of [CS14b] use the Event-B modelling approach [Abr10] to prove the correctness and security of the system. The idea is that the system is described in terms of states it can have and events that transform the state. Event-B uses set theory as a modelling notation and refinement to represent systems at different abstraction levels. Refinement intuitively means that one system implements another. The goal is to verify consistency between different refinement levels.

The security of C-S is proved in a weaker model that has a number of restrictions on the adversarial power. In the following, we study the security of the C-S BB system in the threat model introduced in Section 3.2.

4.2.1 Attacking Confirmable Persistence

Observe that in the Posting protocol users are not allowed to post clashing data. This allows the Soundness property (bb.3) to hold. Persistence property holds as in case of honest WBB, it will never accept inconsistently posted items nor remove any published items.

Confirmable Persistence property considers security against corrupted WBB. All data published must be accompanied with a valid receipt (WBBreceipt), thus

an attack against Stability and Unremovability will be an attack against (t_s, N_c)-TSS and Σ. To be precise, the C-S BB assumes (k, t_s, N_c) - TSS where $t_s = t_c$ and $k = N_c - t_s > t_s + 1$, and claim that their protocol can tolerate $< N_c/3$ corrupted IC peers. However, they do not refer the reader to any specific constructions of TSSs. Schemes with such properties do exist (e.g., [Sho00, AMN01]), but they are based on a trusted dealer. Having a trusted third party should be avoided in our scenario.

TSSs based on distributed interactive key generation protocol have been introduced (e.g., [Bol03]), but they are based on the assumptions that $k = t_s + 1$ and $t_s < N_c/2$, the latter is due to the key generation phase. Therefore, we take $t_s < N_c/2$.

Consider the following attack against Confirmable Persistence and especially Unremovability (property (bb.4)). Assume that we have $(t_s = \lceil N_c/2 \rceil - 1, N_c)$-EUFCMA-secure TSS and $t_c = \lceil N_c/3 \rceil - 1 < N_c/3$. We show that a corrupted WBB can output two valid $\mathsf{WBBreceipt}$s for a period p, i.e.,

$$
\begin{aligned}
w &= \mathsf{WBBreceipt}[p, B_p] = (m = (p, B_p), \sigma = \mathsf{TSign}_{\mathsf{ssk}}(m)),\\
w' &= \mathsf{WBBreceipt}[p, B'_p] = (m' = (p, B'_p), \sigma' = \mathsf{TSign}_{\mathsf{ssk}}(m')),\\
w &\neq w' : \mathsf{TssVf}_{\mathsf{pk}}(m, \sigma) = \mathsf{TssVf}_{\mathsf{pk}}(m', \sigma') = 1.
\end{aligned}
$$

Let $t_h := t_s - t_c$. In period p user U posts an item x, but does not obtain a receipt $\mathsf{rec}[x]$, because the adversary blocked the communication such that $(x, p) \in B_{i,p}$ for only one honest peer P_i. In the end of period p, the Optimistic protocol is executed, and the adversary blocks communication such that exactly $(t_s+1)-t_c = t_h+1 = \lceil N_c/2 \rceil - \lceil N_c/3 \rceil + 1$ honest peers and corrupted peers threshold sign a B_p such that $(x, p) \notin B_p$ and send it to the WBB. More precisely, the WBB receives $(t_h + 1) + t_c = t_s + 1$ signature shares on (p, B_p) such that $x \notin B_p$, and can output a valid receipt w. Remaining honest peers execute the Fallback followed by the Optimistic protocol, and agree on B'_p such that $(x, p) \in B'_p$. After that, the WBB can output a valid receipt w' such that $x \in B'_p$.

Therefore, a corrupted WBB can output two valid w and w' for one period p, and the Unremovability property is violated. In the next section, we show that the C-S BB system tolerates $t_c < N_c/4$ corrupted IC peers to achieve Confirmable Persistence when not using the trivial TSS defined in Section 2.3.

4.2.2 Attacking Confirmable Liveness

Before describing an attack against Confirmable Liveness, we present the liveness assumptions of C-S BB system. One of the following three conditions should hold.

- *All IC peers are honest and online.* No adversary is considered for IC peers.
- *The number of corrupted peers* $< N_c/3$ *and all users are honest.* Under this assumption, all users must always broadcast their postings. As a result all IC peers will be aware of this. We will show that this is crucial to achieve liveness as a corrupted user may engage in the Posting protocol such that liveness is affected.
- *The number of corrupted peers* $< N_c/3$*, but they do not change their database once it is fixed and will not send different databases to different peers, but users can be corrupted.* In this case adversary $\mathcal{A}$ is considered to be covert. This assumption implies that adversary can still cause stopping failures (due to protocol re-runs after malicious detection) which is an attack against liveness.

Culnane and Schneider allow peers to go arbitrarily offline and online as long as always a threshold set of honest peers is online. If that is the case, the adversary can make peers go online and offline so that no messages are never delivered and liveness cannot be achieved. Thus, we do not consider this as an option.

In general, these liveness assumptions are not preserved under a standard security framework. Next, we show that without aforementioned assumptions, the liveness of C-S BB system can be broken.

Consider the following attack. In period p an honest user U_h broadcasts x_h to all peers P_i, $i \in [N_c]$, and obtains a valid receipt $\mathsf{rec}[x_h]$. Next, a corrupted user U_c deviates from broadcasting and sends x_c to all t_c corrupted IC peers and exactly $N - 2t_c$ honest IC peers. Denote the latter set of honest peers by $\mathcal{H}[x_c]$. Observe, that even if t_c honest peers are not even aware of x, the collaboration of $t_c + (N_c - 2t_c) = N_c - t_c$ is enough so that U_c obtains a valid receipt $\mathsf{rec}[x_c]$, yet $x_c \in B_{i,p}$ only for honest peers $\mathsf{P}_i \in \mathcal{H}[x_c]$. Denote by $\mathcal{S}[x_c]$ the t_c honest peers s.t. $x_c \notin B_{i,p}$.

Next, the Optimistic protocol is executed and corrupted peers send their signed local BB records only to honest peers from set $\mathcal{H}[x_c]$. After that step, all corrupted peers remain inert. Therefore, honest peers from set $\mathcal{S}[x_c]$ want to execute the Fallback protocol, but honest peers from set $\mathcal{H}[x_c]$ send their signed databases to the WBB. As a result, the WBB does not receive $N_c - t_c$ signature shares on B_p and cannot publish anything for period p. Hence, liveness cannot be achieved.

The weakness in the design of C-S exploited by the above attack is that users are the only ones responsible for providing the posted items to the IC peers. Thus, liveness cannot be achieved in case there exist corrupted IC peers. In the following, we propose re-broadcasting of posted items to guarantee that if at least one honest peers receives an item, then eventually all honest peers will do so.

4.3 Confirmable Persistence of Culnane-Schneider Bulletin Board System

The authors of [CS14b] use Event-B modelling and refinement approach to prove the correctness and security of their BB system, but its actual cryptographic security is unknown.

In this section, we prove Confirmable Persistence for C-S BB system. Recall that since it has only one WBB peer, we denote $\mathsf{WBB} := \mathsf{WBB}_1$ and the view of the WBB as $L_{\mathsf{pub},T} := L_{\mathsf{pub},1,T}$. Let $\mathsf{Prec}[p]$ be the set of periods preceding p. The total view of the WBB at some T during period p, is the union of the agreed and published BB records for all periods in $\mathsf{Prec}[p]$.

For clarity, in the following proofs for C-S BB system, we do not consider hash functions, we assume that H is collision-resistant.

The verification algorithms $\mathsf{VerifyRec}$ and $\mathsf{VerifyPub}$ are defined as

$$\mathsf{VerifyRec}(\mathsf{rec}[x], x, \mathsf{cr}_{\mathsf{U}}, \mathsf{params}) := \mathsf{TssVf}_{\mathsf{pk}}(m = (p, x), \mathsf{TSign}_{\mathsf{ssk}}(m)),$$
$$\mathsf{VerifyPub}(L_{\mathsf{pub},T}, \mathsf{params}) := \bigwedge_{\tilde{p} \in \mathsf{Prec}[p]} \mathsf{TssVf}_{\mathsf{pk}}(m = (\tilde{p}, B_{\tilde{p}}), \mathsf{TSign}_{\mathsf{ssk}}(m)),$$

where $L_{\mathsf{pub},T} := \bigcup_{\tilde{p} \in \mathsf{Prec}[p]} \Big\{ \mathsf{WBBreceipt}[\tilde{p}, B_{\tilde{p}}] \Big\}$.

We begin with a useful Lemma.

Lemma 4.1. *Assume that* TSS *is* (t_s, N_c)*-EUF-CMA-secure. Assume that* $t_h \geq t_c$*, where* $t_h := t_s - t_c$*. Let* $\mathcal{A}$ *be an arbitrary PPT adversary. If* $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E}) = 1$ *for the C-S BB system (and the BB system from Section 5), then for any period* p *with* $1 - \mathbf{negl}(\kappa)$ *probability there exist a set* $\mathcal{H} := \{\mathsf{P}_{i_k}\}_{k \in [t_h+1]}$ *of honest IC peers that output* $\mathsf{ShareSig}_{\mathsf{ssk}_{i_k}}((p, B_p))$ *for* $k \in [t_h + 1]$.

Proof. Let $T > T_{\mathsf{end},p}$ be a moment after the end of period p. If $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E}) = 1$ then $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T}\big) = \mathsf{accept}$. Therefore, $\mathsf{WBBreceipt}[p, B_p] = \big(m = (p, B_p), \sigma = \mathsf{TSign}_{\mathsf{ssk}}(m)\big) \in L_{\mathsf{pub},T}$ such that $\mathsf{TssVf}_{\mathsf{pk}}(m, \sigma) = 1$.

As a contradiction, assume that less than $t_h + 1$ honest peers output a threshold signature on (p, B_p). We construct the following adversary $\mathcal{A}_{\mathsf{TSS}}$ that breaks the static (t_s, N_c)-EUF-CMA security of TSS.

$\mathcal{A}_{\mathsf{TSS}}$ invokes $\mathcal{A}$ and emulates the game $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E})$ playing the role of the challenger $\mathcal{C}$, as follows: the adversary $\mathcal{A}$ responds with the set of corrupted peers L_{corr}, so $\mathcal{A}_{\mathsf{TSS}}$ in turn sends the set of corrupted IC peers $\mathcal{IC} \subseteq L_{\mathsf{corr}}$ to the (t_s, N_c)-EUF-CMA challenger and hence corrupts the same subset of peers (players). Each time $\mathcal{A}$ makes a signature share query for some $\mathsf{P}_i \notin \mathcal{IC}$, $\mathcal{A}_{\mathsf{TSS}}$ makes the same query to the signing oracle who returns the correct signature share, and forwards the signature share to $\mathcal{A}$.

Since $\mathsf{TssVf}_{\mathsf{pk}}((p, B_p), \sigma) = 1$, and $\mathcal{A}$ queried less than $t_h + 1$ honest peers then $\mathcal{A}$ forged the threshold signature σ. If that is the case, $\mathcal{A}_{\mathsf{TSS}}$ outputs $((p, B_p), \sigma)$ as

a successful forgery. Thus, if $\mathcal{A}$ wins $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E})$ with non-negligible probability, then $\mathcal{A}_{\mathsf{TSS}}$ outputs a successful forgery with non-negligible probability which contradicts to the (t_s, N_c)-EUF-CMA-security of TSS. Therefore, with $1 - \mathbf{negl}(\kappa)$ probability, some set $\mathcal{H}$ of $t_h + 1$ honest peers threshold signed (p, B_p). □

Next, we are ready to give a proof for Confirmable Persistence.

Theorem 4.2. *Let $N_c, t_c, t_s \in \mathbb{N}$. Let Σ be an EUF-CMA-secure signature scheme and TSS be a (t_s, N_c)-EUF-CMA-secure TSS. If $t_c < (t_s + 1)/2$, then the C-S BB system with N_c IC peers over TSS achieves t_c-Confirmable Persistence.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against the Confirmable Persistence game $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E})$. In the following $t_h := t_s - t_c$. We show that with $1 - \mathbf{negl}(\kappa)$ probability neither of the properties (P.1) nor (P.2) from Figure 1 can hold and $\mathsf{VeryfyPub}$ accepts. Consider the following cases.

There is a moment T' such that $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$, then (P.1) can not hold. Suppose there exists an item (x, p), honest peers $\mathcal{H}' = \{\mathsf{P}_{i_k}\}_{k\in[t_c+1]}$ such that $\mathcal{H}' \cap L_{\mathsf{corr}} = \emptyset$, and moments $\{T_{i_k}\}_{k\in[t_c+1]}$ with the following properties. For every moment $T' \geq \min\limits_{k\in[t_c+1]}\{T_{i_k}\}$:

(i) $(x, p) \in L_{\mathsf{pub},T'}$;
(ii) $\forall k \in [t_c + 1]$ $(x, p) \notin L_{\mathsf{post},i_k,T_{i_k}}$;
(iii) $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$.

Take $T' = \min\limits_{k\in[t_c+1]}\{T_{i_k}\}$. As $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$ and TSS is (t_s, N_c)-unforgeable, it follows from Lemma 4.1 that with $1 - \mathbf{negl}(\kappa)$ probability at least $t_h + 1$ honest peers signed B_p such that $(x, p) \in B_p$ no later than moment T'.

An honest peer threshold signs B_p if it has obtained $N_c - t_c$ signatures on (p, B_p). Suppose now that less than $N_c - 2t_c$ of the signatures are from honest peers. We show that in that case we can construct an adversary $\mathcal{A}_\Sigma$ that breaks the unforgeability of Σ.

$\mathcal{A}_\Sigma$ invokes $\mathcal{A}$ and plays $\mathcal{C}$ in the Confirmable Persistence game. $\mathcal{A}_\Sigma$ runs $\mathcal{A}$ that returns L_{corr}, the set of corrupted peers. Challenger for the EUF-CMA game sends pk to $\mathcal{A}_\Sigma$. Adversary $\mathcal{A}_\Sigma$ assigns key pk to a random peer $\mathsf{P}_i \notin L_{\mathsf{corr}}$, and for the rest of the system, runs setup as usual. $\mathcal{A}_\Sigma$ interacts with $\mathcal{A}$ and collects signatures on (p, B_p) to a set S. With certain probability P_i's signature is in S and P_i is the honest peer that did not sign (p, B_p). Adversary $\mathcal{A}_\Sigma$ returns P_i's signature as a forgery.

Therefore, at least $N_c - 2t_c$ of the signatures are from honest peers P_i and for these honest peers $(x, p) \in B_{i,p}$. Since $(N_c - 2t_c) + (t_c + 1) = N_c - t_c + 1 > N_c - t_c$, we have a contradiction.

There exists a moment T' such that $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$*, then (P.2) can not hold.* First, let us show that for a single period p with $1 - \mathbf{negl}(\kappa)$ probability $\mathcal{A}$ can output a valid threshold signature for exactly one database B_p.

Assume to the contrary that $\mathcal{A}$ outputs

$$\begin{aligned} w &= \mathsf{WBBreceipt}[p, B_p] = (m = (p, B_p), \sigma = \mathsf{TSign}_{\mathsf{ssk}}(m)), \\ w' &= \mathsf{WBBreceipt}[p, B'_p] = (m' = (p, B'_p), \sigma' = \mathsf{TSign}_{\mathsf{ssk}}(m')), \\ w &\neq w' : \mathsf{Vf}_{\mathsf{pk}}(m, \sigma) = \mathsf{Vf}_{\mathsf{pk}}(m', \sigma') = 1. \end{aligned}$$

It follows from Lemma 4.1, that with probability $1 - \mathbf{negl}(\kappa)$ some set $\mathcal{H}$ of $t_h + 1$ honest peers threshold signed (p, B_p) and some set $\mathcal{H}'$ of $t_h + 1$ honest peers threshold signed (p, B'_p). According to the protocol description, $\mathcal{H} \cap \mathcal{H}' = \emptyset$.

An honest IC peer threshold signs B_p if it has obtained $N_c - t_c$ signatures on (p, B_p). Suppose now that less than $N_c - 2t_c$ of the signatures are from honest peers. In that case we can construct an adversary $\mathcal{A}_\Sigma$ that breaks EUF-CMA-security of Σ, the construction of $\mathcal{A}_\Sigma$ is the same as above.

Therefore, at least $N_c - 2t_c$ of the signatures are from honest peers. Denote them by set $\mathcal{S}$. Since $|\mathcal{S}| + |\mathcal{H}'| \geq N_c - 2t_c + t_h + 1 > N_c - t_c$ at least one honest peer P_i threshold signs B'_p and also sends signature on B_p. If P_i threshold signs B'_p, then the protocol is over for it and it would not sign B'_p. Then it must first sign B_p and then threshold sign the database B'_p. Therefore $B_p \subseteq B'_p$.

Arguing similarly we get that there must me be some honest peer P_j that first signed B'_p and later threshold signed B_p. Then also $B'_p \subseteq B_p$. Hence, $B_p = B'_p$, which contradicts our assumption.

Next, we show that (P.2) can not hold. We have that $L_{\mathsf{pub},T} := \bigcup_{\tilde{p} \in \mathsf{Prec}[p]} \Big\{\mathsf{WBBreceipt}[\tilde{p}, B_{\tilde{p}}]\Big\}$. Suppose that there exist moments T and $T' > T$, and $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T}\big) = \mathsf{accept}$ such that $(x, \hat{p}) \in L_{\mathsf{pub},T}$ and $(x, \hat{p}) \notin L_{\mathsf{pub},T'}$. As $(x, \hat{p}) \in L_{\mathsf{pub},T}$, it follows that there exists $\hat{p} \in \mathsf{Prec}[p]$ such that $(x, \hat{p}) \in B_{\hat{p}}$. On the contrary, as $(x, \hat{p}) \notin L_{\mathsf{pub},T'}$, we have that $(x, \hat{p}) \notin B'_{\hat{p}}$. Since for every period p adversary can output valid signature for one database, we have a contradiction. □

The above implies that if we have $t_s < N_c/2$, the C-S BB system tolerates roughly $< N_c/4$ corrupted IC peers, Culnane and Schneider claim that their system can tolerate $< N_c/3$ corrupted IC peers. We note that when using the trivial threshold signature defined in Section 2, the protocol tolerates roughly $< N_c/3$ corrupted IC peers.

5 A Fully Secure Bulletin Board System

In this section, we present an enhanced BB system based on the C-S BB system which is fully secure in our framework proposed in Section 3. More precisely, our fully secure BB system achieves (Confirmable) Persistence and Confirmable Liveness for $< N_c/3$ and $< N_c/2$ corrupted IC peers, respectively. Furthermore, the enhanced system is simple from the communication complexity aspect. This system is also introduced in our paper.

5.1 Description of the System

As in Section 4.1, we use a signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$ and a (t_s, N_c)-threshold signature scheme $\mathsf{TSS} = (\mathsf{DistKeygen}, \mathsf{ShareSig}, \mathsf{ShareVerify}, \mathsf{TssVf}, \mathsf{Combine})$. Denote by sk_i the individual signature key of peer P_i. Let ssk be the threshold signature key, and let ssk_i be its ith share.

As the C-S BB system, the enhanced system contains users, IC peers denoted by $\mathsf{P}_i, i \in \{1, \ldots, N_c\}$, and a single WBB. The new system also contains SA that initializes the entities. The protocol runs in periods and consists of the Posting protocol and the Publishing protocol. Each period p is a time interval between $T_{\mathsf{begin},p}$ and $T_{\mathsf{end},p}$, i.e., $p := [T_{\mathsf{begin},p}, T_{\mathsf{end},p}]$. For each period p an IC peer P_i has a local record $B_{i,p}$ of received items x and a local database $D_{i,p}$ of received peers' signatures of items x. In the beginning of each period p, each peer P_i has $B_{i,p}, D_{i,p} \leftarrow \emptyset$, i.e., databases are initialized empty.

We enhance the C-S protocol by re-broadcasting x. Namely, an IC peer P_i will re-broadcast item x, if it has not received it from a user U but received $(j, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_j}(m))$ from another IC peer P_j. This prevents the attack described in Section 4.2.2. Furthermore, the new Publishing protocol does not include Optimistic and Fallback protocols. In the new Publishing protocol, in the end of each period each IC peer (threshold)signs its local BB record and sends it to the WBB.

Assume that we have a period p. The protocol for user U to post item x is as follows.

Posting Protocol:

$\mathsf{U} \rightarrow \mathsf{P}_i$**:** User U broadcasts item an x with credential cr_U to all peers P_i, $i \in [N_c]$.

$\mathsf{P}_i \rightarrow \mathsf{P}_j$**:** Upon receiving x from U, each P_i checks that cr_U is valid for U, i.e $\mathsf{Accept}(\mathsf{U}, x) = 1$. If the check is successful, then it broadcasts $(i, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_i}(m))$ to all other IC peers.

P_i**:** P_i waits for messages $(j, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_j}(m))$ from peers P_j, $j \neq i$, and adds them to the dataset of received signed items $D_{i,p}$.

– Upon receiving a message $(j, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_j}(m))$, if P_i has not received $(x, \mathsf{cr}_\mathsf{U})$ from some user and has not previously broadcast $(i, m, \mathsf{Sig}_{\mathsf{sk}_i}(m))$, and $\mathsf{Accept}(\mathsf{U}, x) = 1$, then it broadcasts $(i, m, \mathsf{Sig}_{\mathsf{sk}_i}(m))$ to all other IC peers.
– Upon receiving $N_c - t_c - 1$ valid signatures on $(p, x, \mathsf{cr}_\mathsf{U})$, P_i adds (p, x) to local BB record $B_{i,p}$ for period p.

$\mathsf{P}_i \to \mathsf{U}$: Upon adding (p, x) to $B_{i,p}$, P_i sends $(i, m = (p, x), \mathsf{ShareSig}_{\mathsf{ssk}_i}(m))$ to U.

U: The user U waits for valid TSS share signatures $(j, m = (p, x), \mathsf{ShareSig}_{\mathsf{ssk}_j}(m))$ from $N_c - t_c \geq t_s + 1$ peers P_j, $j \neq i$. When this happens, let S_U be a set of $N_c - t_c$ IC peers from which U can combine the share. U sets obtains the receipt for x

$$\mathsf{rec}[x] := \mathsf{TSign}_{\mathsf{ssk}}((p, x)) \leftarrow \mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_{N_c}, (p, x), (j, \sigma_j)_{j \in S_\mathsf{U}}) \ .$$

In the end of each period p, local BB records are published. The publishing protocol of our fully secure BB system is presented below.

Publishing Protocol:

$\mathsf{P}_i \to \mathsf{WBB}$: Each P_i sends $(i, m = (p, B_{i,p}), \mathsf{ShareSig}_{\mathsf{ssk}_i}(m))$ to the WBB.
– WBB waits for messages $(j, m = (p, B_{j,p}), \sigma_j = \mathsf{ShareSig}_{\mathsf{ssk}_j}(m))$ from at least $t_s + 1$ peers j.

– Let S be a set of those $\geq N_c - t_c$ peers that agree on the contents of the BB and let B_p be the board of agreed-on contents. Obtain

$$\mathsf{TSign}_{\mathsf{ssk}}((p, B_p)) \leftarrow \mathsf{Combine}(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_{N_c}, (p, B_p), (j, \sigma_j)_{j \in S}) \ .$$

– Publish $\mathsf{WBBreceipt}[p, B_p] := (m = (p, B_p), \mathsf{TSign}_{\mathsf{ssk}}(m))$.

5.2 Confirmable Persistence

We use the same notation as in Section 4.3, recall that $\mathsf{VerifyRec}$ and $\mathsf{VerifyPub}$ are defined as

$$\mathsf{VerifyRec}(\mathsf{rec}[x], x, \mathsf{cr}_U, \mathsf{params}) := \mathsf{TssVf}_{\mathsf{pk}}(m = (p, x), \mathsf{TSign}_{\mathsf{ssk}}(m)),$$
$$\mathsf{VerifyPub}(L_{\mathsf{pub},T}, \mathsf{params}) := \textstyle\bigwedge_{\tilde{p} \in \mathsf{Prec}[p]} \mathsf{TssVf}_{\mathsf{pk}}(m = (\tilde{p}, B_{\tilde{p}}), \mathsf{TSign}_{\mathsf{ssk}}(m)),$$

where $L_{\mathsf{pub},T} := \bigcup_{\tilde{p} \in \mathsf{Prec}[p]} \left\{ \mathsf{WBBreceipt}[\tilde{p}, B_{\tilde{p}}] \right\}$, and $\mathsf{Prec}[p]$ is the set of periods preceding p. In other words, the total view of the WBB at some T during period p, is the union of the agreed and published BB records for all periods in $\mathsf{Prec}[p]$. Next theorems follow the notation from formal definition of (Confirmable) Persistence (see Definition 3.1).

Theorem 5.1 (Confirmable Persistence)**.** *Let $N_c, t_c, t_s \in \mathbb{N}$ and set $t_s = N_c - t_c - 1$. Let* TSS *be a (t_s, N_c)-EUF-CMA-secure TSS (e.g., the trivial TSS). If $t_c < N_c/3$, then the fully secure BB system with N_c IC peers over* TSS *achieves Confirmable Persistence for tolerance threshold t_c.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary against Confirmable Persistence game $\mathcal{G}^{\mathcal{A},\mathcal{P},t_c}_{\mathsf{C.Prst}}(1^\kappa, \mathbf{E})$. We set $t_h := t_s - t_c$. If $t_c < N_c/3$, then $t_h > t_c - 1$. We show that with $1 - \mathsf{negl}(\kappa)$ probability neither of the properties (P.1) nor (P.2) from Figure 1 can hold and VerifyPub accepts. This implies that $\mathcal{A}$ cannot win and completes the proof.

There is a moment T' s.t. if $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}, \mathsf{params}\big) = \mathsf{accept}$*, then (P.1) can not hold:* Suppose there exists an item (x, p), honest peers $\mathcal{H}' = \{\mathsf{P}_{i_k}\}_{k\in[t_c+1]}$ such that $\mathcal{H}' \cap L_{\mathsf{corr}} = \emptyset$, and moments $\{T_{i_k}\}_{k\in[t_c+1]}$ with the following properties. For every moment $T' \geq \min\{T_{i_k} : k \in [t_c + 1]\}$:

(i) $(x, p) \in L_{\mathsf{pub},T'}$;
(ii) $\forall k \in [t_c + 1]$ $(x, p) \notin L_{\mathsf{post},i_k,T_{i_k}}$;
(iii) $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$.

Take $T' = \min\{T_{i_k} : k \in [t_c + 1]\}$. As $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}\big) = \mathsf{accept}$ and TSS is (t_s, N_c)-EUF-CMA-secure, it follows from Lemma 4.1 that at least $t_h + 1$ honest peers signed B_p such that $(x, p) \in B_p$ no later than moment T'. Since $t_h = N_c - 2t_c - 1$, we have that $(t_h + 1) + (t_c + 1) > N_c - t_c$, so there exists an honest peer that threshold signed two databases in period p (one with (x, p) and one without (x, p)). This contradicts the Publishing protocol description.

There is a moment T' s.t. if $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T'}, \mathsf{params}\big) = \mathsf{accept}$*, then(P.2) can not hold:* First, let us show that for a single period p with $1 - \mathbf{negl}(\kappa)$ probability $\mathcal{A}$ can output a valid threshold signature for exactly one database.

Assume to the contrary that $\mathcal{A}$ outputs

$$
\begin{aligned}
w &= \mathsf{WBBreceipt}[p, B_p] = (m = (p, B_p), \sigma = \mathsf{TSign}_{\mathsf{ssk}}(m)),\\
w' &= \mathsf{WBBreceipt}[p, B'_p] = (m' = (p, B'_p), \sigma' = \mathsf{TSign}_{\mathsf{ssk}}(m')),\\
w &\neq w' : \mathsf{Vf}_{\mathsf{pk}}(m, \sigma) = \mathsf{Vf}_{\mathsf{pk}}(m', \sigma') = 1.
\end{aligned}
$$

It follows from Lemma 4.1, that with $1 - \mathsf{negl}(\kappa)$ probability, some set $\mathcal{H}$ of $t_h + 1$ honest peers threshold signed (p, B_p) and some set $\mathcal{H}'$ of $t_h + 1$ honest peers threshold signed (p, B'_p). According to the Publishing protocol description, we have that $\mathcal{H} \cap \mathcal{H}' = \emptyset$. Since $t_c < N_c/3$, we have that $(t_h + 1) + (t_h + 1) = 2(t_s - t_c + 1) = 2(N_c - t_c - 1 - t_c + 1) = 2(N_c - 2t_c) > N_c - t_c$, so at least one honest IC peer threshold signed both B_p and B'_p. This leads to a contradiction as according to the Publishing protocol an honest peer threshold signs only one database in period p.

Now we show that (P.2) can not hold. Recall that we have that $L_{\mathsf{pub},T} := \bigcup_{\tilde{p}\in\mathsf{Prec}[p]} \left\{\mathsf{WBBreceipt}[\tilde{p}, B_{\tilde{p}}]\right\}$. Suppose that there exist moments T and $T' > T$, and $\mathsf{VerifyPub}\big(L_{\mathsf{pub},T}\big) = \mathsf{accept}$ such that $(x, \hat{p}) \in L_{\mathsf{pub},T}$ and $(x, \hat{p}) \notin L_{\mathsf{pub},T'}$. As $(x, \hat{p}) \in L_{\mathsf{pub},T}$, it follows that there exists $\hat{p} \in \mathsf{Prec}[p]$ such that $(x, \hat{p}) \in B_{\hat{p}}$. On the contrary, as $(x, \hat{p}) \notin L_{\mathsf{pub},T'}$, we have that $(x, \hat{p}) \notin B'_{\hat{p}}$. Since for every period p adversary can output valid signature for one database, we have a contradiction. □

Observe that in the fully secure BB system, an honest WBB adds B_p to $L_{\mathsf{pub},T}$ only if it can combine a valid signature $\mathsf{TSign}_{\mathsf{ssk}}((p, B_p))$. Thus, the following Theorem follows trivially.

Theorem 5.2 (Persistence)**.** *Let $N_c, t_c, t_s \in \mathbb{N}$ and set $t_s = N_c - t_c - 1$. Let* TSS *be a (t_s, N_c)-EUF-CMA-secure TSS (e.g., the trivial TSS). If $t_c < N_c/3$, then the fully secure BB system with N_c IC peers over* TSS *achieves Persistence for tolerance thresholds $(t_c, 0)$.*

Our fully secure BB system requires $t_s = N_c - t_c - 1$, i.e., we must use the trivial TSS defined in Section 2.3. This is indeed equal to claiming that instead of TSS, we use EUF-CMA-secure Σ, and require that a user/the WBB waits for $N_c - t_c$ valid signatures. We use the language of TSS for consistency with the C-S BB system.

We recall that in case of our trivial TSS the signature is a concatenation of individual signatures. Thus, a valid threshold signature is longer than for classical TSS. As N_c is rather small for BB systems, this is not an issue.

5.3 Confirmable Liveness

In this section, we give a proof for Confirmable Liveness (see Definition 3.2) of the new fully secure BB system. Recall that Confirmable Liveness is not achieved by the original C-S BB system described in Section 4. In the following proof, we show that for every $p = [T_{\mathsf{begin},p}, T_{\mathsf{end},p}]$, an honest user that posts an item x during p at least some specified time prior to $T_{\mathsf{end},p}$ will obtain a receipt for x and x will be published on the WBB. Taking account the delay message bound δ and synchronization loss bound Δ, we consider that each (honest) peer P_i stops broadcasting messages in the Posting protocol at local time $\mathsf{Clock}[\mathsf{P}_i] = T_{\mathsf{end},p} + \Delta$ and stops receiving messages and engages in the Publishing protocol for period p at local time $\mathsf{Clock}[\mathsf{P}_i] = T_{\mathsf{end},p} + \Delta + \delta$.

Theorem 5.3 (Confirmable θ-Liveness)**.** *Let $\theta, \delta, \Delta, N_c, t_c, t_s \in \mathbb{N}$. Let Σ be an EUF-CMA-secure signature scheme and* TSS *be a (t_s, N_c)-EUF-CMA-secure TSS. Let $T_{\mathsf{comp}} \in \mathbb{N}$ be the time complexity upper bound of the processes in the fully*

secure protocol. If $t_c < t_s + 1 \leq N_c - t_c$*, then the BB system described with* N_c *IC peers over* Σ *and* TSS *achieves* θ*-Confirmable Liveness for fault tolerance thresholds* $(t_c, 0)$*, delay message bound* δ *and synchronization loss bound* Δ*, and for every* $\theta \geq \Delta + 3\delta + (2N_c + 3)T_{\mathsf{comp}}$.

Proof. For $\theta \geq \Delta + 3\delta + (2N_c + 3)T_{\mathsf{comp}}$, consider an adversary $\mathcal{A}$ against the θ-Confirmable Liveness game $\mathcal{G}^{\mathcal{A},\delta,\Delta,t_c,0}_{\theta-\mathsf{C.Live}}(1^\kappa, \mathbf{E})$ that wins. Assume that conditions (L.1) and (L.2) of $\mathcal{G}^{\mathcal{A},\delta,\Delta,t_c,0}_{\theta-\mathsf{C.Live}}(1^\kappa, \mathbf{E})$ hold. Namely,

(L.1) for some honest user U, $\mathcal{A}$ provides $\mathcal{C}$ with the message $(\mathsf{post}, \mathsf{U}, x)$ at global time $\mathsf{Clock} = T$, where T is during a period $p = [T_{\mathsf{begin},p}, T_{\mathsf{end},p}]$.

(L.2) no Publishing protocol execution happens during global time interval $[T, T + \theta]$.

In the BB system, the Publishing protocol starts at global time $\mathsf{Clock} = T_{\mathsf{end},p}$, which suggests that U engaged at the Posting protocol at global time $\mathsf{Clock} = T \leq T_{\mathsf{end},p} - \theta$.

We will show that condition (L.3) can not hold, which implies that $\mathcal{A}$ can not win the game and completes the proof. We analyse the following cases.

(L.3.a) can not hold: By global time $\mathsf{Clock} \leq T + \theta$*,* U *will obtain a valid receipt* $\mathsf{rec}[x]$ *for* x *with* $1 - \mathsf{negl}(\kappa)$ *probability.* Based on the description of the Posting protocol in Section 5.1, we show that the waiting time for U is upper bounded by the value $\theta^* := \Delta + 3\delta + (2N_c + 3)T_{\mathsf{comp}}$. In our computation, we always consider time advancement according to the description of the the Posting protocol and under the restrictions posed in the Confirmable Liveness game, i.e., message delay bound δ and synchronization bound Δ. We assume that if global clock $\mathsf{Clock} \leq T'$, then for all entities X $\mathsf{Clock}[X] \leq T' + \Delta$, and $\mathcal{A}$ cannot change the global clock Clock.

By (L.1), when U broadcasts $(x, \mathsf{cr}_\mathsf{U})$, the global time is $\mathsf{Clock} = T$. Hence, taking into account the clock synchronization bound Δ, each peer P_i has then internal time $\mathsf{Clock}[\mathsf{P}_i] \leq T + \Delta$, and each P_i will receive x at global time time

$$\mathsf{Clock} \leq T + \delta.$$

Upon receiving $(x, \mathsf{cr}_\mathsf{U})$, each P_i checks the validity of cr_U, and computes (all computation is done in time $2 \cdot T_{\mathsf{comp}}$) and broadcasts the value $(i, m = (p, x, \mathsf{cr}_\mathsf{U}), \mathsf{Sig}_{\mathsf{sk}_i}(m))$ to all other IC peers by global time

$$\mathsf{Clock} \leq T + \delta + 2T_{\mathsf{comp}}.$$

Therefore, according to the Posting protocol description, each peer P_i will receive signatures from other honest peers by global time

$$\mathsf{Clock} \leq (T + \delta + 2T_{\mathsf{comp}}) + \delta = T + 2\delta + 2T_{\mathsf{comp}},$$

as user U broadcast $(x, \mathsf{cr}_{\mathsf{U}})$ to all IC peers.

In order for an honest peer P_i to add (p, x) to $B_{i,p}$, it must receive and verify the validity of $N_c - t_c - 1$ signatures from the other peers, as the adversary can also send at most t_c invalid messages on behalf of malicious IC peers[1]. The time required for verification is at most $(t_c + (N_c - t_c - 1)) \cdot T_{\mathsf{comp}} = (N_c - 1) \cdot T_{\mathsf{comp}}$, and the signature share will be created in time T_{comp}. Therefore, each honest P_i will send its TSS share at global time

$$\mathsf{Clock} \leq (T + 2\delta + 2T_{\mathsf{comp}}) + N_c T_{\mathsf{comp}} = T + 2\delta + (N_c + 2)T_{\mathsf{comp}}.$$

The user U will obtain TSS shares from all honest peers by global time

$$\mathsf{Clock} \leq (T + 2\delta + (N_c + 2)T_{\mathsf{comp}}) + \delta = T + 3\delta + (N_c + 2)T_{\mathsf{comp}}.$$

The user U requires at most $N_c \cdot T_{\mathsf{comp}}$ to verify all shares (again, $\mathcal{A}$ can send invalid shares). Finally, the user U will require at most T_{comp} time to combine the TSS shares and obtain her receipt by global time

$$\mathsf{Clock} \leq (T + 3\delta + (N_c + 2)T_{\mathsf{comp}}) + (N_c + 1)T_{\mathsf{comp}} = T + 3\delta + (2N_c + 3)T_{\mathsf{comp}},$$

and by internal time

$$\mathsf{Clock}[\mathsf{U}] \leq T + \Delta + 3\delta + (2N_c + 3)T_{\mathsf{comp}}.$$

Consequently, we set the upper bound θ^* for the waiting time of U as

$$\theta^* := \Delta + 3\delta + (2N_c + 3)T_{\mathsf{comp}}.$$

The validity of the receipt that U computes derives directly from the fact that $N_c - t_c \geq t_s + 1 > t_c$ and TSS is (t_s, N_c)-EUF-CMA-secure with $1 - \mathbf{negl}(\kappa)$ probability.

If there is a moment T' s.t. for every $T'' \geq T'$, x is included in the view of WBB, then (L.3.b) can not hold. To prove this statement, we first show that by the end of period p, all honest peers agree on their local records, i.e., for every distinct honest peers $\mathsf{P}_i, \mathsf{P}_j$, it holds that $B_{i,p} = B_{j,p}$. This happens because the following two cases hold.

(i) All honest peers agree on the actions of the honest users: since condition (L.2) of $\mathcal{G}^{\mathcal{A},\delta,\Delta,t_c,0}_{\theta-\mathsf{C.Live}}(1^\kappa, \mathbf{E})$ and $\theta \geq \theta^*$, every honest peer P_i contributes to the generation of receipts of all honest users by providing its TSS share when receiving an honest post $(x, \mathsf{cr}_{\mathsf{U}})$. Recall that an honest user will always engage in the Posting

[1]The malicious IC peers can send messages arbitrarily. However, for simplicity and without loss of generality for the liveness guarantee, we treat the messages of each malicious peer as a block. Thus, P_i can receive at most t_c malicious messages.

protocol consistently with posting policy $\mathcal{P}$ by never posting non-accepting items. In addition, by the description in Section 5.1, TSS shares are provided only after the honest peers add an honestly posted item in their local records. By the EUF-CMA security of Σ, with $1-\mathbf{negl}(\kappa)$ probability, when verification is complete P_i is ascertained that $N_c - t_c - 1$ honest peers have received and signed x. Therefore, each honest peer P_i includes every honestly posted item x in $B_{i,p}$.
(ii) All honest peers agree on the actions of the malicious users: this case captures of the following two sub-cases:

(a) A malicious user $\hat{\mathsf{U}}$ posted an item $\hat{x}$ such that at least honest peer P_i is aware of $\hat{x}$: by the description in Section 5.1, P_i will broadcast $\hat{x}$ upon receiving it from $\hat{\mathsf{U}}$. Therefore, with some delay, all honest peers will receive $\hat{x}$ and engage in the process of agreement for receipt generation. The latter implies that each honest peer P_j will either reject $\hat{x}$ as P_i will do or accept $\hat{x}$ as P_i will do, resulting in the addition of $\hat{x}$ in every honest local BB record.

(b) A malicious user $\hat{\mathsf{U}}$ posted an item $\hat{x}$ such that no honest IC peer is aware of $\hat{x}$: For instance, $\hat{\mathsf{U}}$ partially broadcast x only to the t_c malicious peers. In this case, clearly no honest IC peer will include $\hat{x}$ in its local record. (Observe that if malicious IC peer sends his signature on $\hat{x}$ to at least one honest IC peer P_i, we can argue similarly as in (a), i.e., if P_i accepts x, with some delay, all honest peers will receive $\hat{x}$ and engage in the process of agreement for receipt generation if P_i accepts.)

Taking into account the message delay δ for re-broadcasting among the honest IC peers, and the fact that each honest P_i stops broadcasting messages in the Posting protocol at time $\mathsf{Clock}[\mathsf{P}_i] = T_{\mathsf{end},p} + \Delta$ and stops receiving messages and engages in the Publishing protocol at time

$$\mathsf{Clock}[\mathsf{P}_i] = T_{\mathsf{end},p} + \Delta + \delta,$$

agreement on the local BB records is guaranteed. As a result, all $N_c - t_c \geq t_s + 1 > t_c$ honest peers will provide the WBB with $(i, m = (p, B_{i,p}), \mathsf{ShareSig}_{\mathsf{ssk}_i}(m))$ by global time at most

$$\mathsf{Clock} \leq (T_{\mathsf{end},p} + \delta) + T_{\mathsf{comp}} = T_{\mathsf{end},p} + \delta + T_{\mathsf{comp}}.$$

In the above we have that an IC peer creates a signature share in time T_{comp}. Then, the WBB will receive all $N_c - t_c$ shares and combine them to produce and publish $\mathsf{WBBreceipt}[p, B_p]$ for period p by global time

$$\begin{aligned}\mathsf{Clock} &\leq (T_{\mathsf{end},p} + \delta + T_{\mathsf{comp}}) + \delta + (N_c + 1)T_{\mathsf{comp}} \\ &= T_{\mathsf{end},p} + 2\delta + (N_c + 2)T_{\mathsf{comp}}.\end{aligned}$$

In the above, we assume that all WBB computation is done in time $(t_c + N_c - t_c) \cdot T_{\mathsf{comp}} + T_{\mathsf{comp}}$ as WBB needs to wait for at least $N_c - t_c$ valid signature shares and

combine a valid TSS. As the adversary can not create more than t_c TSS shares for records that do not include x, and the view of the honest WBB is append only, we have that for global time

$$\mathsf{Clock} := T' = T_{\mathsf{end},p} + \Delta + 2\delta + (N_c + 2)T_{\mathsf{comp}}$$

and for any $T'' \geq T'$, x is included in the view of the WBB. This concludes the proof. □

It follows from Theorem 5.3 that the new fully secure BB system tolerates $< N_c/2$ corrupted IC peers to achieve Confirmable Liveness.

6 Related Work

In this section, we give an overview of other BB systems introduced in the literature. We consider the idea of using Byzantine Agreement protocols and describe BB systems of Peters [Pet05], Heather and Lundin [HL09], Krummenacher [Kru10], STAR-Vote [BBK+12, BBB+13], D-DEMOS [CZZ+16] and Dini [Din03].

We stress that from the above [HL09] focuses only on BB and has been published, STAR-Vote, D-DEMOS and e-voting service proposed by Dini have been indeed published in a conference/journal, but describe entire voting system and do not focus on BB systems.

6.1 Byzantine Agreement Problem

The problem of achieving consensus in a distributed system in case of Byzantine, i.e., arbitrary failures, is known as the Byzantine Agreement (BA) problem [LSP82]. The idea is that all honest peers must agree on the output value and it can be shown that these protocols tolerate strictly less that $N/3$ faulty peers.

The authors of [CGS97] claim that a BB system can be viewed as a secure broadcast channel and it can be implemented as a set of replicated servers implementing a BA protocol. Classical BA protocols can be used to agree on the vote-set, but they are not suitable for implementing vote collection. Furthermore, the majority of such protocols require synchronous communication model and have unreasonable computation complexity for practical use in the voting scenario.

In general, Byzantine fault-tolerance is an important characteristic of a system and a number of Byzantine fault-tolerant algorithms using state machine replication have been introduced and implemented, e.g., [CL02] and [SB12]. These algorithms tolerate $< N/3$ faulty replicas, but they are much more costly and their actual cryptographic security is unknown, and must be thoroughly studied before applying it to e-voting scenario.

6.2 Peters' Bulletin Board

In his Master's thesis Peters [Pet05] studies different protocols implementing a secure broadcast channel. He compares communication and computation complexity, threat models and chooses to implement a variation of a protocol proposed in [Rei94] on top of a secure group membership protocol from [Rei96]. He states that his goal is to give an implementation and he does not concentrate on studying the cryptographic security and complete descriptions for the protocols.

The overall idea is somewhat similar to a two-phase commit protocol, i.e., a user sends a message to a random IC peer. This particular peer first broadcasts a so called init message and others reply with signed echo message. Once the peer

receives signatures from more than $2N_c/3$ peers, it broadcasts a commit message. Again a peer waits for more than $2N_c/3$ signed confirmations, constructs the receipt and sends it to the user.

In order to improve the communication and computation complexity, Peters replaces digital signatures with threshold signatures. Peters does not assume the existence of any trusted third party and stresses that to get a completely distributed BB no trusted set-up must be required. The set-up phase is completely distributed and, therefore, the threshold signature scheme tolerates only $t_s < N/2$ and by definition $t_c + 1$ signature shares are required to generate a valid threshold signature. In his work, Peters considers [Bol03] to be a suitable signature scheme. As the cryptographic security of the system is not studied, it might be an issue. This can be solved using the trivial TSS. In his Master's thesis [Beu12], Beuchat implements a BB based on [Pet05], but uses a TSS of [Sho00] that also requires a trusted third party for key generation.

In the BB system, the secure group membership protocol [Rei96] is used to maintain the set of currently operational and correct peers.

Peters work includes a detailed description of the protocols and justifies the correctness in this setting, but without formal security analysis and verification this system cannot be used in a real world setting. Furthermore, Peters assumes the network is always reliable (not only for liveness), i.e., all honest peers can always communicate with each other and an honest peer failing to communicate is considered to be corrupted and will lead to execution of the secure group membership protocol.

Therefore, the threat model and lack of formal cryptographic proof makes this system not suitable for use in real-world election process, at least before it has not been analysed cryptographically.

6.3 Heather and Lundin's Bulletin Board

Heather and Lundin [HL09] propose a BB system motivated by the requirements of e-voting systems. Heather and Lundin identify the following security properties: Unalterable History which in an analogue of Unremovability (bb.4); Certified Publishing which is an analogue of Stability (bb.1); Timely Publication which is somewhat similar to Confirmable Liveness, meaning that a posted item must eventually be published.

They propose a system containing only a single peer and prove that it satisfies the aforementioned requirements. The system uses locking and hashing, thus, it does not scale well. Heather and Lundin also point out the importance of a distributed and fault-tolerant BB. They briefly discuss possible approaches, but leave concrete system and its security analysis for future work.

6.4 Krummenacher's Bulletin Board

In MSE Seminar on E-Voting Krummenacher [Kru10] proposes a distributed BB system for e-voting applications inspired by the system introduced in [HL09].

Krummenacher improves the approach of Heather and Lundin making it fault-tolerant and distributed using threshold signatures. In his implementation, a BB is public, i.e., everyone can read published items, once an item is published it cannot be removed or deleted, and each reader must be able to detect insertion, change and deletion.

Similarly the approach of Heather and Lundin, Krummenacher makes very strong assumptions, e.g., he assumes that there exists an adequate public key infrastructure. Krummenacher states that the system is distributed, but uses RSA threshold signature [Sho00], which has a trusted dealer. (To overcome this issue, the trivial threshold signature scheme can be used.)

There are both synchronous and asynchronous versions of the protocol. However, it is stated that the synchronous version is not suitable for e-voting applications, because it cannot tolerate large amount of write requests due to locking. Thus, in the following, we concentrate on the asynchronous version.

Different variants of asynchronous protocols and their disadvantages are presented. In one version, the user publishes on several boards in parallel leading to loss of performance by locking, but it is claimed that once a user receives a receipt it is guaranteed that l out of N_c IC peers have published the message, where l is the size of the threshold set, i.e., TSS is used to create the receipt.

In another variant of the asynchronous protocol, the user posts a message independently to l out of N_c peers, in this case it is not guaranteed that all l peers will also publish the message as threshold signing is not used. In both cases, IC peers may have their own history of items posted and to get a full public WBB, these histories must be combined.

Furthermore, Krummenacher describes a concrete implementation of BB, but does not present a formal security proof. Krummenacher states that the system is correct if more than $N_c/2$ of IC peers are up and running. We stress that (in comparison to C-S or the fully secure BB system) the usage of locks leads to loss of performance.

6.5 Bulletin Board of STAR-Vote

STAR-Vote ([BBK+12, BBB+13]) is an example of a kiosk-based voting system which is designed to tolerate faulty peers. Although the functionality of the BB in STAR-Vote is also to collect the votes and publish them in the end of an election period, the setting is completely different from remote e-voting. In other words, in STAR-Vote, voting terminals play the role of BB. As a result, the collection

of votes is not addressed and only the problem of tracking the counted votes is considered.

This is claimed to be solved by using the approach of [SDW08]. The idea is that Voting terminals in polling stations maintain a log of votes using a hash chain, i.e., votes are simply hashed into the chain and, therefore, the log can not be altered. This cannot be applied in a remote e-voting setting when the system must scale and handle multiple users posting items. Furthermore, there is no formal protocol description nor security proof, and the threat model is very weak, i.e., local network is considered to be reliable.

In conclusion, STAR-Vote includes a protocol for collecting votes and posting them to a remote WBB, the voting terminals just maintain a log of cotes using a hash chain. However, it does not address the problem of inconsistencies between the voting terminals.

6.6 Bulletin Board of D-DEMOS

D-DEMOS is a distributed I-voting system introduced by Chrondros et al. in [CZZ+16], which addresses the issues of DEMOS [KZZ15]. In DEMOS vote vollection is done by a centralized Election Authority, but in D-DEMOS BB functionality is divided into two distributed sub-systems, namely, the vote collection sub-system (in our notation, IC sub-system) and the BB sub-system (in our notation, the WBB sub-system). The IC sub-system is responsible for vote collection and runs two protocols: the Voting protocol and the Vote-Set Consensus protocol. Both protocols are explicitly described and analysed. The protocol can tolerate $< N_c/3$ corrupted IC peers and $< N_c/2$ corrupted WBB peers.

The idea is that a user sends a vote only to one IC peer which is responsible for delivering the receipt back to the user. This IC peer communicates with other IC peers to create a receipt and a uniqueness certificate for the ballot indicating that this user has voted making use of verifiable threshold secret sharing and digital signatures. In the end of election period, IC sub-system runs the Vote-Set Consensus protocol to guarantee that all honest IC peers agree on the vote-set. After the execution of the latter protocol, IC nodes publish their results to all WBB peers. A WBB peers accepts a vote-set when it receives $t_c + 1$ identical sets.

The main difference with [CS14b] is that the vote collection is completely asynchronous. Furthermore, the set of votes is uploaded to the distributed WBB in the end of election period. We believe that it is due to the complexity of the Vote-Set Consensus protocol, because for each vote ballot Bracha's binary consensus protocol [Bra87] is executed. We do not consider it computationally reasonable in real-world elections to perform a binary consensus for each ballot. We are interested in an approach where votes can be published in some period p so that voters get the opportunity to validate their votes. However, in D-DEMOS vote can be

validated only in the end of the election period.

The security of D-DEMOS is studied in a cryptographic model. More precisely, the authors of [CZZ+16] show that if a voter obtains a receipt, then with high probability it is assured that the vote will be published on the honest WBB peer. Furthermore, the liveness is achieved in asynchronous communication model. We note that in our framework presented in Section 3, we make use of the notation from [CZZ+16].

6.7 E-Voting Service of Dini

Dini [Din03] proposes a distributed e-voting service following the approach of [FOO92]. It is based on replication and tolerates arbitrary failures. Dini does not focus on the BB system, but on an e-voting service in general.

Dini requires the system to achieve Eligibility, Uniqueness, Privacy and Availability. The drawbacks of the approach are that a sender anonymous channel is needed and that the voter will have to run a validation protocol. On the other hand, in our approach, we formalize the requirements of voting where it is assumed that the voter gets a credential that authenticates her during ballot casting with the voting collection system, therefore, no anonymous channel requirements are needed. Furthermore, Dini assumes the existence of reliable channels in his security model.

7 Conclusion

In this thesis, we studied BB systems for e-voting from functionality and security aspect. Based on the analysis of security requirements proposed by Culnane and Schneider in Computer Security Foundations Symposium 2014, we introduced a cryptographic framework for secure BB systems. We expanded the model of Garay et al. from Eurocrypt 2015 for secure blockchain protocols, and considered a secure BB as an RPTL that additionally supports the generation of receipts for successful postings. We described two properties: Persistence and Confirmable Liveness. We used confirmability in the Liveness property to capture the generation of receipts. We also extended the Persistence property to be confirmable. In other words, the dishonesty of WBB can be detected via verification of published data.

We analysed the security of C-S BB system. We showed that C-S BB does not achieve Confirmable Persistence for $< N/3$ corrupted IC peers without using the trivial threshold signature scheme. In case of using a TSS with distributed key generation, the bound is $< N/4$. Furthermore, we attacked liveness of C-S BB and showed that it does not achieve Confirmable Liveness in case of corrupted IC peers.

Inspired by the security analysis of C-S BB, we presented a fully secure and simple BB system that tolerates $< N/3$ corrupted IC peers for Confirmable Persistence and $< N/2$ corrupted IC peers for Confirmable Liveness.

This thesis is based on a submitted paper "A Cryptographic Approach to Bulletin Boards" with co-authors Aggelos Kiayas, Helger Lipmaa, Janno Siim and Thomas Zacharias. We note that security proofs in this thesis are more detailed compared to the submitted paper. Furthermore, in our paper we do not prove Confirmable Persistence for C-S BB, and do not present related work on BB systems as thoroughly as in this thesis.

In future work, the concept of clashing items must be further studied. Namely, in Estonian e-voting, it is required that as a countermeasure against corruption it must be possible to re-vote. Currently all votes are published on the WBB which can not be the case for that scenario. Furthermore, the proposed fully secure BB system needs to be implemented and benchmarked.

References

[Abr10] Jean-Raymond Abrial. *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[AL10] Yonatan Aumann and Yehuda Lindell. Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries. *J. Cryptology*, 23(2):281–343, 2010.

[AMN01] Michel Abdalla, Sara K. Miner, and Chanathip Namprempre. Forward-Secure Threshold Signature Schemes. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 441–456, San Francisco, CA, USA, April 8–12, 2001. Springer, Heidelberg.

[BBB+13] Susan Bell, Josh Benaloh, Michael D. Byrne, Dana Debeauvoir, Bryce Eakin, Philip Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, Dan S. Wallach, Gail Fisher, Julian Montoya, Michelle Parker, and Michael Winn. STAR-Vote: A Secure, Transparent, Auditable, and Reliable Voting System. In *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*, Washington, D.C., 2013. USENIX Association.

[BBK+12] Josh Benaloh, Mike Byrne, Philip T. Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, and Dan S. Wallach. STAR-Vote: A Secure, Transparent, Auditable, and Reliable Voting System. *CoRR*, abs/1211.1904, 2012.

[BCS16] Craig Burton, Chris Culnane, and Steve Schneider. vVote: Verifiable Electronic Voting in Practice. *IEEE Security and Privacy*, 14(4):64–73, 2016.

[Beu12] Jose Beuchat. Realization of a Secure Distributed Bulletin Board, Master's Thesis. Bern University of Applied Sciences, 2012.

[Bol03] Alexandra Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46, Miami, Florida, USA, January 6–8, 2003. Springer, Heidelberg.

[Bra87] Gabriel Bracha. Asynchronous Byzantine Agreement Protocols. *Inf. Comput.*, 75(2):130–143, November 1987.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. *A Secure and Optimally Efficient Multi-Authority Election Scheme*, pages 103–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[Cha81] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha01] David Chaum. Surevote: International patent WO 01/55940 A1, 2001.

[CL02] Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.*, 20(4):398–461, November 2002.

[CRST15] Chris Culnane, Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. vVote: A Verifiable Voting System. *ACM Trans. Inf. Syst. Secur.*, 18(1):3:1–3:30, 2015.

[CS14a] Chris Culnane and Steve Schneider. A Peered Bulletin Board for Robust Use in Verifiable Voting Systems. *CoRR*, abs/1401.4151, 2014.

[CS14b] Chris Culnane and Steve A. Schneider. A Peered Bulletin Board for Robust Use in Verifiable Voting Systems. In *CSF 2014*, pages 169–183, Vienna, Austria, July 19–22, 2014. IEEE Computer Society.

[CZZ+16] Nikos Chondros, Bingsheng Zhang, Thomas Zacharias, Panos Diamantopoulos, Stathis Maneas, Christos Patsonakis, Alex Delis, Aggelos Kiayias, and Mema Roussopoulos. D-DEMOS: A Distributed, End-to-End Verifiable, Internet Voting System. In *ICDCS 2016*, pages 711–720, Nara, Japan, June 27–30, 2016. IEEE Computer Society.

[Dam88] Ivan Bjerre Damgård. *Collision Free Hash Functions and Public Key Signature Schemes*, pages 203–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.

[Des88] Yvo Desmedt. Society and Group Oriented Cryptography: A New Concept. In *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, CRYPTO '87, pages 120–127, London, UK, UK, 1988. Springer-Verlag.

[Din03] Gianluca Dini. A Secure and Available Electronic Voting Service for a Large-Scale Distributed system. *Future Generation Computer Systems*, 19(1):69–85, 2003.

[FL16] Prastudy Fauzi and Helger Lipmaa. Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 200–216, San Franscisco, CA, USA, February 29–March 4, 2016. Springer, Heildeberg.

[FLZ16] Prastudy Fauzi, Helger Lipmaa, and Michał Zajac. A Shuffle Argument Secure in the Generic Model. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016 (2)*, volume 10032 of *LNCS*, pages 841–872, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *AUSCRYPT*, pages 244–251, 1992.

[Fur05] Jun Furukawa. Efficient and Verifiable Shuffling and Shuffle-Decryption. *IEICE Transactions*, 88-A(1):172–188, 2005.

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. In *EUROCRYPT*, pages 281–310, 2015.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.*, 17:281–308, 1988.

[HL09] James Heather and David Lundin. The Append-Only Web Bulletin Board. In Pierpaolo Degano, Joshua Guttman, and Fabio Martinelli, editors, *Formal Aspects in Security and Trust: 5th International Workshop, FAST 2008 Malaga, Spain, October 9-10, 2008 Revised Selected Papers*, pages 242–256, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[HT94] Vassos Hadzilacos and Sam Toueg. A Modular Approach to Fault-Tolerant Broadcasts and Related Problems. Technical report, 1994.

[HW14] Sven Heiberg and Jan Willemson. Verifiable Internet Voting in Estonia. In *EVOTE 2014*, pages 1–8, Oct 2014.

[Kru10] Roland Krummenacher. Implementation of a Web Wulletin Board for E-Voting Applications. *MSE Seminar on E-Voting. Institute for Internet Technologies and Applications*, 2010.

[KZZ15] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. *End-to-End Verifiable Elections in the Standard Model*, pages 468–498. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[Lam77] Leslie Lamport. Proving the Correctness of Multiprocess Programs. *IEEE Trans. Softw. Eng.*, 3(2):125–143, March 1977.

[LJY14] Benoît Libert, Marc Joye, and Moti Yung. Born and Raised Distributively: Fully Distributed Non-Interactive Adaptively-Secure Threshold Signatures with Short Shares. In *PODC 2014*, pages 303–312, Paris, France, July 15–18, 2014. ACM Press.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. *The Byzantine Generals Problem*, volume 4, pages 382–401. ACM, New York, NY, USA, July 1982.

[LZ12] Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. In Ivan Visconti and Roberto De Prisco, editors, *SCN 2012*, volume 7485 of *LNCS*, pages 477–502, Amalfi, Italy, September 5–7, 2012. Springer, Heidelberg.

[Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, http://bitcoin.org/bitcoin.pdf, 2008.

[Nef01] C. Andrew Neff. A Verifiable Secret Shuffle and Its Application to E-Voting. In *ACM CCS 2001*, pages 116–125, Philadelphia, Pennsylvania, USA, November 6–8 2001. ACM Press.

[Pet05] R. A. Peters. A Secure Bulletin Board, Master's Thesis. Eindhoven University of Technology, 2005.

[Rei94] Michael K. Reiter. Secure Agreement Protocols: Reliable and Atomic Group Multicast in Rampart. In *Proceedings of the 2Nd ACM Conference on Computer and Communications Security*, CCS '94, pages 68–80, New York, NY, USA, 1994. ACM.

[Rei96] Michael K. Reiter. A Secure Group Membership Protocol. *IEEE Trans. Softw. Eng.*, 22(1):31–42, January 1996.

[SB12] João Sousa and Alysson Neves Bessani. From Byzantine Consensus to BFT State Machine Replication: A Latency-Optimal Transformation. In Cristian Constantinescu and Miguel P. Correia, editors, *EDCC*, pages 37–48. IEEE Computer Society, 2012.

[SDW08] Daniel Sandler, Kyle Derr, and Dan S. Wallach. VoteBox: A Tamper-Evident, Verifiable Electronic Voting System. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 349–364. USENIX Association, 2008.

[Sha79] Adi Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sho00] Victor Shoup. Practical Threshold Signatures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 207–220, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg.

[TW10] Björn Terelius and Douglas Wikström. Proofs of Restricted Shuffles. In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT 2010*, volume 6055 of *LNCS*, pages 100–113, Stellenbosch, South Africa, May 3–6, 2010. Springer, Heidelberg.

Non-exclusive licence to reproduce thesis and make thesis public

I, Annabell Kuldmaa (date of birth: 17th of July 1993),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

On Secure Bulletin Boards for E-Voting

supervised by Helger Lipmaa

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 18.05.2017