Unconditionally Secure Signature Schemes Revisited

Colleen M. Swanson and Douglas R. Stinson*

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
c2swanso,dstinson@uwaterloo.ca

Abstract. Unconditionally secure signature (USS) schemes provide the ability to electronically sign documents without the reliance on computational assumptions needed in traditional digital signatures. Unlike digital signatures, USS schemes require both different signing and different verification algorithms for each user in the system. Thus, any viable security definition for a USS scheme must carefully treat the subject of what constitutes a valid signature. That is, it is important to distinguish between signatures that are created using a user's signing algorithm and signatures that may satisfy one or more user verification algorithms. Moreover, given that each verifier has his own distinct verification algorithm, a USS scheme must necessarily handle the event of a disagreement. In this paper, we present a new security model for USS schemes that incorporates these notions, as well as give a formal treatment of dispute resolution and the trust assumptions required. We provide formal definitions of non-repudiation and transferability in the context of dispute resolution, and give sufficient conditions for a USS scheme to satisfy these properties. Finally, we give an analysis of the construction of Hanaoka et al. in our security model.

1 Introduction

Unconditionally secure signature (USS) schemes provide the ability to electronically sign documents without the reliance on computational assumptions needed in traditional digital signatures. That is, USS schemes are the analogue of digital signatures in the unconditionally secure cryptographic setting. The construction of such schemes is interesting not only from a theoretical perspective, but also from the viewpoint of ensuring security of information in the long term or designing schemes that are viable in a post-quantum world.

Unlike digital signatures, USS schemes require both different signing and different verification algorithms for each user in the system. Thus, any viable security definition for a USS scheme must carefully treat the subject of what constitutes a valid signature. That is, it is important to distinguish between signatures that are created using a user's signing algorithm and signatures that may satisfy one or more user verification algorithms. Current research [5, 6, 10, 12, 7] has proposed various models for unconditionally secure signature schemes, but these models do not fully treat the implications of having multiple verification algorithms or analyze the need for (and trust questions associated with) having a dispute resolution mechanism. We address both of these issues in this paper.

Historically, there have been several attempts to create unconditionally secure constructions that satisfy security properties required for digital signatures, including non-repudiation, transferability, and unforgeability. Chaum and Roijakkers [2] introduced unconditionally secure signatures, proposing an interactive scheme that does not have transferability. Another approach to creating unconditionally secure signatures has been to enhance existing unconditionally secure message authentication codes (MACs), making these codes more robust in a signature setting.

* Research supported by NSERC grant 203114-06

MACs clearly do not provide non-repudiation, as the sender and receiver compute authentication tags using the same algorithm. In addition, the need for a designated sender and receiver further limits the applicability of such schemes to a general signature setting.

Much research has been devoted to the removal of the standard MAC trust assumptions, in which both sender and receiver are assumed to be honest. In A^2-codes [13, 14, 8], the sender and receiver may be dishonest, but there is a trusted arbiter to resolve disputes; in A^3-codes [1, 3, 9], the arbiter is no longer trusted prior to dispute resolution, but is trusted to make an honest decision in event of a disagreement. Johansson [9] used A^3-codes to improve the construction of Chaum and Roijakkers by making it non-interactive, but the signatures produced by the scheme are not transferable, as the use of a designated receiver limits the verification of the signature to those who have the appropriate key. Multi-receiver authentication codes (MRAs) [4] and multi-receiver authentication codes with dynamic sender (DMRAs) [11] use a broadcast setting to relax the requirement for designation of receivers, and also, in the latter case, senders. These codes are not appropriate outside of a broadcast setting, however, as neither non-repudiation nor transferability are satisfied.

Unsurprisingly, the first security models for unconditionally secure signature schemes, including Johansson [9] and Hanaoka et al. [5, 6], drew upon the standard MAC security models. Shikata et al. [12] introduced a model using notions from public-key cryptography, which was also adopted in the work by Hara et al. [7] on blind signatures. Safavi-Naini et al. [10] presented a MAC-based model meant to encompass the notions developed by Shikata et al. In this work, we present a new security model. Our model is more general than the MAC-based models of Hanaoka et al. [5, 6] and Safavi-Naini et al. [10] and covers the attacks described in these works. Like that of Shikata et al. [12], our work is based on security notions from traditional public-key signature systems. However, our model differs from those in the existing literature in its careful treatment of the concept of a "valid" signature. Our aim is to provide a rigorous and natural security model that covers all reasonable attacks.

In addition, we analyze a construction of Hanaoka et al. [5] in our model and provide a proof of security. We remark that while Hanaoka et al. make claims about the security of this construction in their model, they do not provide an analysis. In fact, security proofs are not provided for most of the constructions given in existing research. Thus, we feel it is useful to include our analysis of a basic unconditionally secure signature construction in our security model.

Our basic notion of security is easily extendable to a system with dispute resolution, which we argue is a necessary component of any USS scheme. Furthermore, our treatment of dispute resolution allows us to give formal definitions of non-repudiation and transferability. We show that a USS scheme that satisfies our unforgeability definition and has an appropriate dispute resolution method also satisfies non-repudiation and transferability, both of which are required properties for any reasonable signature scheme. Finally, we define various dispute resolution methods and examine the amount of trust each requires.

An outline of our paper is as follows. In Section 2, we give a basic definition of a USS scheme, before moving to an informal treatment of the desired security properties. We then define a formal security model in Section 3. We formally discuss dispute resolution in Section 4 and give examples of dispute resolution methods in Section 5. In Section 6, we compare our work with that of previous literature. Finally, we analyze the construction of Hanaoka et al. [5] in Section 7 and give some concluding remarks in Section 8.

2 Preliminaries

We require the following definitions.

Definition 2.1. An unconditionally secure signature scheme (or USS scheme) Π consists of a tuple $(\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfying the following:

- The set $\mathcal{U} = \{U_1, \ldots U_n\}$ consists of possible users, X is a finite set of possible messages, and Σ is a finite set of possible signatures.
- The *key-generation algorithm* Gen takes as input a security parameter 1^k and outputs the signing algorithm Sign and the verification algorithm Vrfy. The parameter k is relevant to the overall security of the scheme, as discussed later.
- The *signing algorithm* $\mathsf{Sign}\colon \mathcal{X} \times \mathcal{U} \to \Sigma$ takes a message $x \in \mathcal{X}$ and a signer $U_i \in \mathcal{U}$ as input, and outputs a signature $\sigma \in \Sigma$. For each $U_i \in \mathcal{U}$, we let Sign_i denote the algorithm $\mathsf{Sign}(\cdot, U_i)$.
- The *verification algorithm* $\mathsf{Vrfy}\colon \mathcal{X} \times \Sigma \times \mathcal{U} \times \mathcal{U} \to \{\mathit{True}, \mathit{False}\}$ takes as input a message $x \in \mathcal{X}$, a signature $\sigma \in \Sigma$, a signer $U_i \in \mathcal{U}$, and a verifier $U_j \in \mathcal{U}$, and outputs either *True* or *False*. For each user U_j, we let Vrfy_j denote the algorithm $\mathsf{Vrfy}(\cdot, \cdot, \cdot, U_j)$.

It is required that, for every k, for every pair $(\mathsf{Sign}, \mathsf{Vrfy})$ output by $\mathsf{Gen}(1^k)$, for every pair $U_i, U_j \in \mathcal{U}$, and for every $x \in \mathcal{X}$, it holds that

$$\mathsf{Vrfy}_j(x, \mathsf{Sign}_i(x), U_i) = \mathit{True}.$$

Remark 2.1. We are treating *deterministic* signature schemes only, in the sense that Sign and Vrfy are deterministic, although the above definition can easily be extended to the randomized setting.

We now define the concepts of authentic, acceptable, and fraudulent signatures. Distinguishing these three concepts is one of the main themes of this paper.

Definition 2.2. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is *i-authentic* if $\sigma = \mathsf{Sign}_i(x)$.

Definition 2.3. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is *(i,j)-acceptable* if $\mathsf{Vrfy}_j(x, \sigma, U_i) = T$.

Definition 2.4. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is *(i,j)-fraudulent* if σ is (i,j)-acceptable but not i-authentic.

2.1 Security Notions

Informally, a secure signature scheme should satisfy the following three properties:

1. *Unforgeability*: Except with negligible probability, it should not be possible for an adversary to create a "valid" signature.
2. *Non-repudiation*: Except with negligible probability, a signer should be unable to repudiate a legitimate signature that he has created.
3. *Transferability*: If a verifier accepts a signature, he can be confident that any other verifier will also accept it.

One objective of this paper is to formalize these notions in the unconditionally secure setting; we provide precise definitions in Sections 3 and 4. In contrast to the usual public-key setting, the requirements of non-repudiation and transferability are not guaranteed in a USS scheme that satisfies the above intuitive notion of unforgeability. For "ordinary" digital signatures, non-repudiation is a consequence of unforgeability: a signature is considered "valid" if it passes a verification test, and it should be impossible for anyone to create such a signature without knowledge of the secret signing algorithm. Thus, assuming the signing algorithm is not known to some third party, the signer cannot create a signature and later repudiate it. Transferability of digital signatures is guaranteed since there is a single, public verification algorithm.

In USS schemes, the concept of a "valid" signature requires clarification. A verifier is always capable of finding a signature that passes his own, secret verification test, so we cannot define the validity of a signature based on whether it passes a given user's verification algorithm. Indeed, there must be signatures that pass a given user's verification algorithm but that could not have been created with the signer's signing algorithm; otherwise the scheme will not satisfy unforgeability. Similarly, each verifier's verification algorithm must be different, or a given verifier will be able to present a signature acceptable to any verifier who possesses the same algorithm. A "valid" signature, then, must be created using the signer's signing algorithm, and it should be impossible for anyone to create a signature that *appears* valid to other, non-colluding users, or the scheme will not have the properties of unforgeability, non-repudiation, and transferability. In particular, we have the following observations.

Theorem 2.1. *A necessary condition for a USS scheme to satisfy unforgeability is the existence of (i, j)-fraudulent signatures for $i \neq j$.*

Proof. A verifier U_j can always use his verification algorithm to create an (i, j)-acceptable signature for any $i \neq j$. If there are no (i, j)-fraudulent signatures, then all signatures produced in this fashion must be i-authentic, and therefore they are successful forgeries. □

Theorem 2.2. *A USS scheme must satisfy $\mathsf{Vrfy}_j(\cdot,\cdot,\cdot) \neq \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$ for $j \neq \ell$.*

Proof. Suppose that $\mathsf{Vrfy}_j(\cdot,\cdot,\cdot) = \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$ where $j \neq \ell$. Clearly U_j can create an (i, j)-acceptable signed message, (x, σ). Because $\mathsf{Vrfy}_j(\cdot,\cdot,\cdot) = \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$, it follows immediately that (x, σ) is (i, ℓ)-acceptable. This implies that the user U_ℓ will accept (x, σ) as a valid signature, but (x, σ) was not created by U_i. □

3 Formal Security Model

We now develop a formal security model for USS schemes. Our security definition is comparable to the notion of signatures secure against existential forgery under adaptive chosen message attacks in the case of public-key signature schemes. However, our definition takes into account the peculiarities of the unconditional security setting, in particular the existence of (and need for) fraudulent signatures and multiple verification algorithms.

We specify two types of existential forgery. In our setting, an "existential" forgery is either an (i, j)-fraudulent signature created without the help of the verifier U_j, or an i-authentic signature created without the help of the signer U_i. If a USS scheme is secure, then both of these types of forgeries should be infeasible for an adversary to create.

We need the following oracles:

– The $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ *oracle*; this oracle takes as input a message x and outputs an ℓ-authentic signature for the message x.
– The $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,\cdot)$ *oracle*; this oracle takes as input a signature pair (x,σ) and a signer U_i, and runs user U_ℓ's verification algorithm on input (x,σ,U_i), outputting *True* or *False*.

Definition 3.1. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k, let the set $C \subseteq \mathcal{U}$ be a coalition of at most t users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* Sig-forge$_{C,\Pi}(k)$ with target signer U_i and verifier U_j:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.
2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,U_i)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$. We let $\mathcal{Q}$ denote the set of messages that the coalition submitted as queries to the oracles $\mathsf{Sign}_i^{\mathcal{O}}(\cdot)$. Note that $\mathcal{Q}$ does not contain messages submitted as queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for $\ell \neq i$.
3. The coalition C outputs a signature pair (x,σ) satisfying $x \notin \mathcal{Q}$.
4. The output of the game is defined to be 1 if and only if one of the following conditions is met:
   (a) $U_j \notin C$ and σ is an (i,j)-fraudulent signature on x; or
   (b) $U_i \notin C$ and σ is an i-authentic signature on x.

Definition 3.2. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k and let $\epsilon(k)$ be a negligible function of k. We say Π is $(t, \psi_S, \psi_V, \epsilon)$-*unforgeable* if for all coalitions C of at most t possibly colluding users, and all choices of target signer U_i and verifier U_j,

$$\Pr[\text{Sig-forge}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

Remark 3.1. Another option is to include a $\mathsf{Fraud}_{(i,j)}^{\mathcal{O}}(\cdot)$ *oracle*; this oracle takes as input a message x and outputs an (i,j)-fraudulent signature on x. Providing certain (i,j)-fraudulent signatures to the adversary could only increase his chances of ultimately constructing a new (i,j)-fraudulent signature. Thus this would constitute a stronger security model than the one we consider. On the other hand, it is hard to envisage a scenario where an adversary would have this kind of additional information about a verifier whom the adversary is attempting to deceive. Therefore we do not include the $\mathsf{Fraud}^{\mathcal{O}}$ oracle in our basic model of USS schemes. However, it would be straightforward to modify our model to include these oracles, if desired.

Remark 3.2. We can also define the notion of *strongly unforgeable* USS schemes by appropriately redefining the set $\mathcal{Q}$ of Definition 3.1. That is, we let $\mathcal{Q}$ contain signature pairs of the form (x,σ), where the message x was submitted as a query to the given oracles and the signature σ was the oracle response, and require that the submitted signature pair $(x,\sigma) \notin \mathcal{Q}$.

We observe that a scheme meeting the unforgeability requirement of Definition 3.2 satisfies our intuitive notions of non-repudiation and transferability. We explain these relationships in the following observations, noting that formal definitions of non-repudiation and transferability are intrinsically linked to the dispute resolution process, and so will be provided later, in Section 4. We formalize these observations in Theorems 4.1 and 4.2.

Observation 3.1 *A $(t, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme Π provides non-repudiation.*

Proof. Suppose that Π is $(t, \psi_S, \psi_V, \epsilon)$-unforgeable. Then U_i cannot repudiate a given i-authentic signature σ, as Definition 3.2 guarantees that σ can be created without U_i only with negligible probability (as Condition 4b of Definition 3.1 holds only with negligible probability). Thus U_i cannot claim that other users may have created σ. The other possibility for a signer U_i to repudiate a signature on a message given to U_j is if the signature is (i, j)-fraudulent. Definition 3.2 also implies that U_i cannot create an (i, j)-fraudulent signature (even with the help of $t-1$ other users not including U_j) except with negligible probability, as Condition 4a of Definition 3.1 is assumed to not hold (except with negligible probability). □

Observation 3.2 *A $(t, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme Π provides transferability.*

Proof. In order for a signature σ to be non-transferable from U_j to U_ℓ, σ would have to be (i, j)-acceptable, but not (i, ℓ)-acceptable, where $j \neq \ell$. If σ were i-authentic, it would also be (i, ℓ)-acceptable. Therefore σ must be (i, j)-fraudulent. However, Definition 3.2 implies an (i, j)-fraudulent signature cannot be created without the assistance of U_j, except with negligible probability. □

From the point of view of a verifier, a scheme meeting Definition 3.2 gives reasonable assurance of the validity of a received signature. If a verifier U_j receives a signature pair (x, σ) purportedly from U_i, then U_j will accept the signature so long as σ is (i, j)-acceptable for the message x. In this case, there are only two possibilities: either σ is i-authentic or (i, j)-fraudulent for the message x. If σ is i-authentic, then a coalition that does not include the signer U_i has only a negligible probability of creating σ by Condition 4b of Definition 3.1. If σ is (i, j)-fraudulent, then Condition 4a of Definition 3.1 guarantees that a coalition that does not include U_j cannot create σ, except with negligible probability.

4 Dispute Resolution

Given that each verifier has his own distinct verification algorithm, a USS scheme must necessarily handle the event of a disagreement. That is, since there is no public verification method as in traditional digital signatures, a USS scheme must have a mechanism to determine the authenticity of a signature when some subset of users disagree whether a given signature should be accepted. In particular, dispute resolution is necessary to convince an outsider of the authenticity of a disputed signature. In traditional digital signatures, there are no outsiders to the scheme, in the sense that everyone has access to the public verification method. In our setting, however, the number of participants (and thereby access to verification algorithms) is limited. Dispute resolution is a method that effectively deals with need for resolution of disagreements in, for example, a court setting. Typically, dispute resolution involves all the users voting on the validity of a signature, or alternatively, a trusted arbiter stating whether a signature is valid.

We now incorporate a mechanism for dispute resolution into the basic USS scheme defined in Section 2. We first consider the requirements of a dispute resolution system. With a definition of dispute resolution in place, we can formally define non-repudiation and transferability and give sufficient conditions for a USS scheme to satisfy these properties.

Ideally, the dispute resolution process validates a signature if and only if the signature is authentic, i.e., the signature was produced by the signer. This leads to the following definitions.

Definition 4.1. A *dispute resolution method* $\mathcal{DR}$ for a USS scheme Π is a procedure invoked when a user U_ℓ questions the validity of a given signature (x, σ), purportedly signed by U_i. Here

U_ℓ may be any user in $\mathcal{U}$, including U_i. The procedure $\mathcal{DR}$ consists of an algorithm DR that takes as input a signature pair (x, σ) and a signer U_i, and outputs a value in $\{valid, invalid\}$, together with the following rules:

1. If DR outputs *valid*, then (x, σ) must be accepted as an i-authentic signature on x by all users.
2. If DR outputs *invalid*, then (x, σ) must be rejected by all users.

We remark that the algorithm DR may have access to additional (secret) scheme information, as specified by the particular dispute resolution method.

The following definitions formalize the notion of utility of a given $\mathcal{DR}$.

Definition 4.2. *Soundness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say $\mathcal{DR}$ is *sound* if, whenever σ is not an i-authentic signature on x, then $\mathsf{DR}((x, \sigma), U_i)$ outputs *invalid*.

Definition 4.3. *Completeness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say $\mathcal{DR}$ is *complete* if, whenever σ is an i-authentic signature on x, then $\mathsf{DR}((x, \sigma), U_i)$ outputs *valid*.

Definition 4.4. *Correctness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. If $\mathcal{DR}$ is both sound and complete, we say $\mathcal{DR}$ is *correct*.

With the addition of a dispute resolution method $\mathcal{DR}$, we adjust the unforgeability requirement of a USS scheme by requiring $\mathcal{DR}$ to be sound. Similarly, we require $\mathcal{DR}$ to be performed honestly, in the sense that the adversary is not allowed to modify the algorithm DR or its outputs, as this is a necessary condition for a $\mathcal{DR}$ to be sound (or, in fact, complete). In particular, we recognize a new type of forgery introduced by the dispute resolution process, which necessitates the soundness property of $\mathcal{DR}$:

Definition 4.5. Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say a signature σ on a message x is an *arbiter-enabled forgery* if σ is not i-authentic, but the dispute resolution method $\mathcal{DR}$ outputs *valid* on input σ.

This leads to the following new definition of unforgeability:

Definition 4.6. Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say Π is *$\mathcal{DR}$-unforgeable* with parameters $(t, \psi_S, \psi_V, \epsilon)$ if Π is $(t, \psi_S, \psi_V, \epsilon)$-unforgeable (as in Definition 3.2) and the dispute resolution method $\mathcal{DR}$ is sound.

We now move to a discussion of the properties of non-repudiation and transferability. As previously mentioned, both of these properties are intrinsically linked to the dispute resolution method. That is, the outcome of the dispute resolution method determines the success or failure of these attacks. In particular, we show that completeness is required to achieve both non-repudiation and transferability.

We remark that in order for the dispute resolution method to be invoked in the first place, there must be disagreement as to the validity of a given signature σ. In a repudiation attack, the dispute resolution method is necessarily invoked, as the attack relies on the signer U_i giving a *seemingly* valid signature σ to the verifier U_j and then later denying the validity of σ. Similarly, for a transferability attack, a signature σ that appears valid to U_j is transferred to and rejected by another user U_ℓ, so the dispute resolution method is again invoked. We now provide formal definitions of these two attacks.

Definition 4.7. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let the set $C \subseteq \mathcal{U}$ be a coalition of at most t users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* $\text{Repudiation}_{C,\Pi}(k)$ with signer $U_i \in C$ and target verifier U_j satisfying $U_j \notin C$:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.
2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot, \cdot, U_i)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$.
3. The coalition C outputs a signature pair (x, σ).
4. The output of the game is defined to be 1 if and only if the following conditions are met:
   (a) σ is (i, j)-acceptable, and
   (b) the dispute resolution method $\mathcal{DR}$ rejects σ as invalid.

Definition 4.8. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let $\epsilon(k)$ be a negligible function of k. We say the combined scheme $(\Pi, \mathcal{DR})$ satisfies *non-repudiation* with parameters $(t, \psi_S, \psi_V, \epsilon)$ if for all coalitions C of at most t possibly colluding users, and for all choices of signer U_i and target verifier U_j,

$$\Pr[\text{Repudiation}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

Theorem 4.1. *Let Π be a $(t, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme and let $\mathcal{DR}$ be a complete dispute resolution method for Π. Then $(\Pi, \mathcal{DR})$ provides non-repudiation, provided that $\mathcal{DR}$ is performed honestly.*

Proof. Assume Π does not provide non-repudiation; that is, the game $\text{Repudiation}_{C,\Pi}(k)$ outputs 1 with non-negligible probability. Suppose $\text{Repudiation}_{C,\Pi}(k)$ with signer U_i and target verifier U_j outputs 1. Then C has created an (i, j)-acceptable signature pair (x, σ), such that the dispute resolution method rejects σ as invalid.

Now, σ is either i-authentic or (i, j)-fraudulent. If σ is (i, j)-fraudulent, then Condition 4a of Definition 3.1 holds, so the output of $\text{Sig-forge}_{C,\Pi}(k)$ with target signer U_i and verifier U_j is 1. That is, Π is not $(t, \psi_S, \psi_V, \epsilon)$-unforgeable. If σ is i-authentic, then the dispute resolution method rejected an i-authentic signature and is therefore not complete. □

Definition 4.9. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let the set $C \subseteq \mathcal{U}$ be a coalition of at most t users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* $\text{Non-transfer}_{C,\Pi}(k)$ with signer U_i and target verifier U_j, where $U_j \notin C$:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.
2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot, \cdot, U_i)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$. We let $\mathcal{Q}$ denote the set of messages that the coalition submitted as queries to the oracle $\mathsf{Sign}_i^{\mathcal{O}}(\cdot)$. Note that $\mathcal{Q}$ does not contain messages submitted as queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for $\ell \neq i$.
3. The coalition C outputs a signature pair (x, σ) satisfying $x \notin \mathcal{Q}$.

4. The output of the game is defined to be 1 if and only if the following conditions are met:
   (a) σ is (i,j)-acceptable but not (i,ℓ)-acceptable for some verifier $U_\ell \notin C$; or σ is (i,j)-acceptable and some verifier $U_\ell \in C$ invokes the dispute resolution method $\mathcal{DR}$ (regardless of whether σ is (i,ℓ)-acceptable).
   (b) the dispute resolution method $\mathcal{DR}$ rejects σ as invalid.

Definition 4.10. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter 1^k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let $\epsilon(k)$ be a negligible function of k. We say the combined scheme $(\Pi, \mathcal{DR})$ satisfies *transferability* with parameters $(t, \psi_S, \psi_V, \epsilon)$ if for all choices of signer U_i and target verifier U_j,

$$\Pr[\text{Non-transfer}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

Theorem 4.2. *Let Π be a $(t, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme and let $\mathcal{DR}$ be a complete dispute resolution method for Π. Then $(\Pi, \mathcal{DR})$ provides transferability, provided that $\mathcal{DR}$ is performed honestly.*

Proof. Suppose Π does not provide transferability, and assume the game $\text{Non-transfer}_{C,\Pi}(k)$ outputs 1, with signer U_i and target verifier $U_j \notin C$. Then C output a signature pair (x, σ) such that $x \notin \mathcal{Q}$, σ is (i,j)-acceptable, and the dispute resolution method rejected σ as invalid.

Now, if σ is not (i,ℓ)-acceptable for some U_ℓ, then σ must be (i,j)-fraudulent. This implies that Condition 4a of Definition 3.1 is met. That is, the output of $\text{Sig-forge}_{C,\Pi}(k)$ with target signer U_i and verifier U_j is 1, so Π is not $(t, \psi_S, \psi_V, \epsilon)$-unforgeable.

If σ is not (i,j)-fraudulent (and therefore i-authentic), then the dispute resolution method rejected an i-authentic signature and is therefore not complete. □

Together, Definition 4.6 and Theorems 4.1 and 4.2 outline requirements for a USS scheme Π and a dispute resolution method $\mathcal{DR}$ to satisfy the desired properties of unforgeability, non-repudiation, and transferability. In particular, Π must be $(t, \psi_S, \psi_V, \epsilon)$-unforgeable and $\mathcal{DR}$ must be correct (under the assumption that the adversary is not allowed to modify the algorithm DR).

5 Some Examples of Dispute Resolution Processes

We define three dispute resolution methods and examine the level of trust required in each scheme.

Definition 5.1. We have the following dispute resolution methods, assuming a disputed signature σ on message x with signer U_i:

- *Omniscient Arbiter (OA) Dispute Resolution*: Designate an arbiter equipped with all of the USS scheme set-up information. The signature σ is considered valid if the arbiter, using his knowledge of all the signing and verification algorithms, accepts the signature as authentic.
- *Verifier-equivalent Arbiter (VEA) Dispute Resolution*: Designate an arbiter equipped with his or her own verification algorithm, $\mathsf{Vrfy}_{\mathcal{A}}$, (i.e., the arbiter will be termed a *glorified verifier*). The arbiter tests the authenticity of the signature σ by running $\mathsf{Vrfy}_{\mathcal{A}}(x, \sigma, U_i)$; the signature is considered valid if $\mathsf{Vrfy}_{\mathcal{A}}(x, \sigma, U_i)$ outputs *True*.
- *Majority Vote (MV) Dispute Resolution*: Resolve disputes by having the verifiers vote on the validity of the signature σ. Each verifier is responsible for running his verification algorithm on (x, σ, U_i) and casting a *valid vote* if the verification algorithm outputs *True* and an *invalid vote* otherwise. The signature is considered valid if a predefined threshold of *valid* votes are cast; here we consider the case of a majority threshold and assume all verifiers vote.

However we choose to define the dispute resolution method, it is necessary to determine the amount of trust placed in the arbiter(s) and incorporate this notion into the security model. In particular, we must consider the correctness of these dispute resolution methods.

In the case of OA dispute resolution, we must completely trust the arbiter, as he has all the necessary information to sign and verify documents on behalf of other users. That is, a USS scheme Π with OA dispute resolution clearly cannot satisfy Definition 4.6 unless the arbiter is honest. Moreover, provided that the arbiter is honest, this dispute resolution method is both sound and complete, as the arbiter will be able to determine the authenticity of a given signature and behave appropriately.

In MV and VEA dispute resolution, we can once again achieve correctness by assuming the complete honesty of a majority of verifiers or, respectively, the arbiter. Achieving soundness and completeness is not as clear if we weaken this trust requirement, however. Suppose we establish VEA dispute resolution and we allow the arbiter to be a colluding member of a given coalition; we will argue that soundness is no longer guaranteed.

In the typical VEA setup of current literature [7, 10, 12], the arbiter is assumed to be a glorified verifier, with the same type of keying information as an arbitrary verifier. The arbiter is assumed to follow the rules of the dispute resolution method honestly and is otherwise treated as a normal verifier in the context of the security model, i.e., he is allowed to be dishonest otherwise. We refer to this set of trust assumptions as *standard trust assumptions*.

We argue that the arbiter's distinct role in the dispute resolution method necessitates a more careful study of the arbiter, and that treating the arbiter as a normal verifier in the context of the security model is insufficient. While certainly an arbiter that is dishonest during dispute resolution can cause a fraudulent signature to be deemed valid, we cannot allow the arbiter to be dishonest before dispute resolution either, contrary to the claims of [10, 12]. The case of MV may be viewed as a generalized version of VEA dispute resolution and the security results are similar.

In the following theorem, we demonstrate the existence of an arbiter-enabled forgery in the VEA and MV dispute resolution methods, if we assume that the arbiter(s) may be dishonest prior to dispute resolution. Thus these methods do not achieve soundness under the standard trust assumptions.

Theorem 5.1. *Let Π be a USS scheme and let $\mathcal{DR}$ be a VEA (respectively, MV) dispute resolution method for Π. Suppose Π is $\mathcal{DR}$-unforgeable. Then the arbiter $\mathcal{A}$ is not a member of C (respectively, a majority of verifiers are not in C).*

Proof. In both cases, we assume the dispute resolution process itself is performed honestly, as otherwise Π clearly fails to have sound dispute resolution. (For MV dispute resolution, it suffices to assume the dispute resolution process is performed honestly by a majority of the verifiers.)

We proceed with VEA dispute resolution. By definition, any $(i, \mathcal{A})$-acceptable signature will be accepted by the dispute resolution method. In particular, this implies any $(i, \mathcal{A})$-fraudulent signature will be accepted by the dispute resolution method. If $\mathcal{A} \in C$, then C can create a signature σ on a message x which is $(i, \mathcal{A})$-fraudulent. This signature σ is not i-authentic, but would be accepted by the dispute resolution method, thereby violating soundness.

Similarly, in the case of MV dispute resolution, a group C of dishonest verifiers can create a signature σ on a message x such that σ is (i, ℓ)-fraudulent for any $U_\ell \in C$. If C contains a majority of verifiers, the signature σ would pass the dispute resolution process and be declared valid, thereby violating soundness. □

Theorem 5.1 indicates that a cheating arbiter $\mathcal{A}$ (respectively, a collusion of a majority of verifiers) can successfully forge an (i, j)-fraudulent signature that will be accepted by the dispute resolution method for any cooperating user U_j. Hence, VEA and MV dispute resolution do not protect the signer against dishonest arbiters, since arbiter-enabled forgeries exist.

We remark that completeness in the VEA and MV methods is guaranteed, provided that the dispute resolution process itself is performed honestly. Thus, by Theorem 4.1, a $(t, \psi_S, \psi_V, \epsilon)$-USS scheme Π with VEA or MV dispute resolution provides non-repudiation under the standard trust assumptions. Transferability, as noted in Theorem 4.2, also follows under the standard trust assumptions.

That is, the VEA and MV methods do not require trust in the arbiter(s) prior to dispute resolution in order to achieve non-repudiation and transferability. As seen above, however, the VEA and MV methods do require the arbiter(s) to be honest prior to dispute resolution in order to achieve soundness. In this sense, we see that VEA and MV dispute resolution provide similar *verifier* security to trusted OA dispute resolution, but fail to provide similar *signer* security.

6 Comparison with Existing Models

Our model differs from those in the existing literature in its careful treatment of i-authentic and (i, j)-fraudulent signatures. In comparison to other works, our approach is most similar to that of Shikata et al. [12], whose model is also designed as an extension of traditional public-key signature security notions. We compare our model with [12] in Section 6.1.

The Hara et al. [7] model for unconditionally secure blind signatures is essentially the same as the Shikata et al. model with an added blindness condition. Hara et al. separate the unforgeability definition of [12] into a weaker notion of unforgeability and an additional non-repudiation requirement. The non-repudiation requirement actually treats more cases than a simple non-repudiation attack (as the success of the attack is not dependent on dispute resolution), so the reason for this separation is unclear. The authors of [7] also allow the signer to be the target verifier, which was not explicitly allowed in the Shikata et al. model, and so add a separate unforgeability definition for this case.

The models of Hanaoka et al. [5, 6] and Safavi-Naini et al. [10] are based on security notions from message authentication codes (MACs). Hanaoka et al. treat only a limited attack scenario (which is covered by our model), including *impersonation*, *substitution*, and *transfer with a trap*, and do not include a verification oracle. Safavi-Naini et al. treat a similar range of attacks as our model, specified through *denial*, *spoofing*, and *framing* attacks, and allow both signature and verification oracles. It is unclear whether Safavi-Naini et al. meant to ensure strong unforgeability, as the relationship between successful forgeries and oracle queries is unspecified. Furthermore, our model is more concise, as the denial attack covers a signer trying to repudiate a signature, whereas we show that it is unnecessary to treat non-repudiation as a separate part of an unforgeability definition. In addition, not all attack scenarios included in our definition are covered by the Safavi-Naini et al. model. For instance, the attack consisting of signer $U_i \in C$ with target verifier U_j, where C creates an (i, j)-fraudulent signature, is not considered. The Safavi-Naini et al. model considers this scenario only in the case where an arbiter is involved and rejects the signature (i.e. a denial attack). In certain applications (e.g., e-cash) we do not want the signer to be able to create an (i, j)-fraudulent signature, regardless of whether a dispute resolution mechanism is invoked.

6.1 Comparison with the Model of Shikata et al.

In this section, we discuss several aspects of the model of Shikata et al. [12] and how our approach differs from theirs.

1. The model in [12] is limited to a single-signer scenario. We consider a more general model in which any participant can be a signer.
2. In Definition 2 of [12], a signed message (x, σ) is defined to be *valid* if it was created using the signer's signing algorithm. Then, in their "Requirement 1," which includes notions for verifiability, dispute resolution, and unforgeability, it is stated that (x, σ) is valid if and only if U_j's verification algorithm outputs *True* when given (x, σ) as input. This requirement is problematic, since U_j can use knowledge of his verification algorithm to find a pair (x, σ) that has output *True*; such a pair is then "valid." However, this means that a receiver can create valid signatures, and consequently the signature scheme does not provide unforgeability. Shikata et al. relax this condition in Requirement 2 by allowing a small error probability that an "invalid" signature will be accepted by a given verifier. However, this does not rectify the aforementioned problem, as the probability space in this definition is unspecified.
3. The definitions of *existential forgery* and *existential acceptance forgery* (Definitions 3 and 4, respectively) are rather complicated. It seems that the notion of "existential forgery" corresponds to our definition of an *i-authentic signature*. The coalition that creates this signature should not include U_i. The notion of "existential acceptance forgery" apparently is dependent upon the coalition that creates it. If U_i is in the coalition, then an existential acceptance forgery would most naturally coincide with our definition of an *(i, j)-fraudulent signature*. If U_i is not in the coalition, then it would more likely mean an *(i, j)-acceptable signature*. In each case, the coalition creating the signature should not include U_j. These definitions are a bit confusing, and we believe that the concepts of authentic, acceptable, and fraudulent signatures are helpful in phrasing clear and concise definitions.
4. In Theorem 2 of [12], it is stated without proof that a signature scheme that is "existentially acceptance unforgeable" is necessarily "existentially unforgeable." Roughly speaking, this is logically equivalent to the statement that an adversary that can create an existential forgery can also create an existential acceptance forgery. This statement seems rather obvious, but we need to also consider the coalitions that are creating these signatures. The adversary creating the existential forgery (i.e., an i-authentic signature) could be any coalition C that does not include U_i. An i-authentic signature is an existential acceptance forgery for any user $U_j \notin C \cup \{U_i\}$. However, a problem arises if C consists of all users except for U_i. In this situation, an i-authentic signature created by C is not an existential acceptance forgery for any user. This situation is not accounted for in Theorem 2 of [12], and therefore it does not suffice to consider only existential acceptance forgeries. We remark that our approach is consistent with A^2-codes [14], in which neither the sender nor the receiver is trusted, and so attacks solely against a target signer are considered. Namely, Simmons treats R_0 attacks, impersonation by the receiver, and R_1 attacks, substitution by the receiver. Allowing attacks in which all verifiers collude against a target signer is a generalization of this approach.
5. Notwithstanding the previous points, the definition of "strong security" in [12] (Definition 9) is very similar to our properties 4a and 4b of Definition 3.1, except that Definition 9 only covers existential acceptance forgeries. In order to compare our model with [12], we consider the following three attack scenarios, where U_i denotes the signer and U_j denotes a verifier:
   case A Neither U_i nor U_j is in the coalition C, and C creates an (i, j)-fraudulent signature.
   case B U_i is not in the coalition C, and C creates an i-authentic signature.

case C $U_i \in C$, $U_j \notin C$, and C creates an (i,j)-fraudulent signature.
In our security definition (Definition 3.1), property 4a is equivalent to the union of case A and case C, and property 4b is equivalent to case B. Now, Definition 9 in [12] considers two attacks: property 1) is the union of cases A and B, but does not include the case where there is no target verifier, as discussed in the previous point; and property 2) is case C.

6. Finally, we give a more complete treatment of dispute resolution than is presented in [12].

7 Construction

Current literature favors constructions using multivariate polynomials. We consider the security of the construction from Hanaoka et al. [5] in our security model.

7.1 General Scheme Outline

Key Pair Generation Let $\mathbb{F}_q$ be a finite field with q elements such that $q \geq n$. The TA picks $v_1, \ldots, v_n \in \mathbb{F}_q^\omega$ uniformly at random for users $U_1, \ldots U_n$, respectively. For technical reasons, we assume the n elements $v_1, \ldots v_n \in \mathbb{F}_q^\omega$ satisfy the additional property that for any subset of size $\omega+1$, the corresponding subset of size $\omega+1$ formed from the new vectors $[1, v_1], \ldots, [1, v_n] \in \mathbb{F}_q^{\omega+1}$ is a linearly independent set.

The TA constructs the polynomial $F(x, y_1, \ldots, y_\omega, z)$ as

$$F(x, y_1, \ldots, y_\omega, z) = \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} a_{i0k} x^i z^k + \sum_{i=0}^{n-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} a_{ijk} x^i y_j z^k,$$

where the coefficients $a_{ijk} \in \mathbb{F}_q$ are chosen uniformly at random.

For each user U_ζ for $1 \leq \zeta \leq n$, the TA computes the signing key $s_\zeta(y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega, z)$ and the verification key $\tilde{v}_\zeta(x, z) = F(x, v_\zeta, z)$. It is assumed the TA can communicate with the users via secure channels and deletes the information afterwards.

Signature Generation and Verification For a message $m \in \mathbb{F}_q$, U_ζ generates a signature σ by

$$\sigma(y_1, \ldots, y_\omega) = s_\zeta(y_1, \ldots, y_\omega, m).$$

To verify a signature pair (m, σ) from U_ζ, a user U_ν checks that

$$\sigma(v_\nu) = \tilde{v}_\nu(U_\zeta, m).$$

7.2 Security Results

We consider the game Sig-forge$_{C,\Pi}(k)$ and calculate the probability that the output is 1. In particular, we consider the probability that the coalition C produces a signature pair (x, σ) satisfying Conditions 4a and 4b of Definition 3.1 separately. Here we set $t = \omega$ and $\psi_S = (n-\omega)\psi$, where ψ is the total number of $\mathsf{Sign}_\ell^{\mathcal{O}}$ oracle queries for each user $U_\ell \notin C$. That is, we allow C to have at most ω members and to have access to ψ sample signatures from each user $U_\ell \notin C$. In addition, C has access to ψ_F $\mathsf{Vrfy}^{\mathcal{O}}$ queries.

Theorem 7.1. *Under the above assumptions, C outputs a signature pair (x, σ) in the game* Sig-forge$_{C,\Pi}(k)$ *of Definition 3.1 satisfying Condition 4a with probability at most $\frac{1}{q-\psi_F-1}$ and Condition 4b with probability at most $\frac{1}{q-\psi_F}$.*

Proof. We provide the proof in Appendix A. □

8 Conclusion

We have presented a new security model for unconditionally secure signature schemes, one which fully treats the implications of having multiple verification algorithms. In particular, we have given a formal discussion of dispute resolution, a necessary component of any USS scheme, and analyzed the effect of dispute resolution on unforgeability. We have provided formal definitions of non-repudiation and transferability, and given sufficient conditions for a USS scheme to satisfy these properties. Moreover, we have analyzed the trust assumptions required in typical examples of dispute resolution. Finally, we have given an analysis of the construction of Hanaoka et al. [5] in our security model.

References

1. Brickell, E. and Stinson, D.: Authentication Codes with Multiple Arbiters. In: Eurocrypt '88. LNCS, vol. 330, pp. 51–55, Springer-Verlag (1988)
2. Chaum, D. and Roijakkers, S.: Unconditionally Secure Digital Signatures. In: Crypto '90. LNCS, vol. 537, pp. 206–214, Springer-Verlag (1991)
3. Desmedt, Y. and Yung, M.: Arbitrated Unconditionally Secure Authentication Can Be Unconditionally Protected against Arbiter's Attacks. In: Crypto '90. LNCS, vol. 537, pp 177–188, Springer-Verlag (1990)
4. Desmedt, Y., Frankl, Y., and Yung, M.: Multi-receiver / Multi-sender Network Security: Efficient Authenticated Multicast / Feedback. In: INFOCOM '92, pp. 2045–2054 (1992)
5. Hanaoka, G., Shikata, J., Zheng, Y., and Imai, H.: Unconditionally Secure Digital Signature Schemes Admitting Transferability. In: Asiacrypt 2000. LNCS, vol. 1976, pp. 130–142, Springer (2000)
6. Hanaoka, G., Shikata, J., Zheng, Y., and Imai, H.: Efficient and Unconditionally Secure Digital Signatures and a Security Analysis of a Multireceiver Authentication Code. In: PKC 2002. LNCS, vol. 2274, pp. 64–79, Springer (2002)
7. Hara, Y., Seito, T., Shikata, J., and Matsumoto, T.: Unconditionally Secure Blind Signatures. In: ICITS 2007. LNCS, vol. 4883, pp. 23–43, Springer (2009)
8. Johansson, T.: On the Construction of Perfect Authentication Codes that Permit Arbitration. In: Crypto '93. LNCS, vol. 773., pp. 343–354, Springer-Verlag (1994)
9. Johansson, T.: Further Results on Asymmetric Authentication Schemes. Information and Computation, vol. 151, pp. 100–133 (1999)
10. Safavi-Naini, R., McAven, L., and Yung, M.: General Group Authentication Codes and Their Relation to "Unconditionally-Secure Signatures". In: PKC 2004. LNCS, vol. 2947, pp. 231–247, Springer (2004)
11. Safavi-Naini, R., and Wang, H.: Broadcast Authentication in Group Communication. In Asiacrypt '99. LNCS, vol. 1716, pp.399–412 (2004)
12. Shikata, J., Hanaoka, G., Zheng, Y., and Imai, H.: Security Notions for Unconditionally Secure Signature Schemes. In: Eurocrypt 2002. LNCS, vol. 2332, pp. 434–449, Springer (2002)
13. Simmons, G. J.: Message Authentication with Arbitration of Transmitter/Receiver Disputes. In: Eurocrypt '87. LNCS, pp. 151–165, Springer-Verlag (1988)
14. Simmons, G.: A Cartesian Product Construction for Unconditionally Secure Authentication Codes that Permit Arbitration. J. Cryptology, vol. 2, pp. 77–104 (1990)

A Analysis of Construction

We need the following lemmas:

Lemma A.1. *Let H be a polynomial in $y_1, \ldots, y_\omega$ of the form $\sum_{i=1}^{\omega} a_i y_i$. Suppose H is zero on a set S of ω linearly independent vectors. Then H is the zero polynomial.*

Proof. Since the vectors of S form a basis for the domain of H, we can write any vector in the domain as a linear combination of the elements of S. Since H is a multilinear polynomial in the y_i's with no cross terms, H must be the zero polynomial. □

Lemma A.2. *Let $n \in \mathbb{N}$ and consider the set of $n+1$ vectors*

$$R = \{r_i = (r_{i,1}, \ldots, r_{i,n}) \in \mathbb{F}_q^n : i = 1, \ldots, n+1\}.$$

If the set of vectors $\{r_i' = (1, r_{i,1}, \ldots, r_{i,n}) \in \mathbb{F}_q^{n+1} : i = 1, \ldots, n+1\}$ form a linearly independent set, then there exists a subset $R' \in R$ of linearly independent vectors of size n.

Proof. Consider the matrix

$$M = \begin{pmatrix} 1 & r_{1,1} & r_{1,2} & \ldots & r_{1,n} \\ 1 & r_{2,1} & r_{2,2} & \ldots & r_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & r_{n+1,1} & r_{n+1,2} & & r_{n+1,n} \end{pmatrix}.$$

Let M_{ij} denote the (i,j) minor matrix of M. Then calculating the determinant of M by expansion along the first column, we have

$$\det(M) = \sum_{i=1}^{n+1} (-1)^{i+1} \det(M_{i,1}). \tag{1}$$

Recall M is invertible, so $\det(M) \neq 0$. Thus (1) implies $\det(M_{k,1}) \neq 0$ for some $k \in \{1, \ldots n\}$. We conclude that the matrix $M_{k,1}$ is invertible, so the desired subset R' exists. □

Summary of Coalition's Information

Assume our adversaries are $C = \{U_1, \ldots, U_\omega\}$, with target signer U_ζ and target verifier U_ν.

Set

$$A_j = \begin{pmatrix} a_{0j0} & a_{0j1} & \cdots & a_{0j\psi} \\ a_{1j0} & a_{1j1} & \cdots & a_{1j\psi} \\ \vdots & \vdots & & \cdots \\ a_{(n-1)j0} & a_{(n-1)j1} & \cdots & a_{(n-1)j\psi} \end{pmatrix},$$

where $0 \leq j \leq \omega$,

and write

$$F(x, y_1, \ldots, y_\omega, z) = \begin{pmatrix} 1 & x & \cdots & x^{n-1} \end{pmatrix} (A_0 + y_1 A_1 + \ldots y_\omega A_\omega) \begin{pmatrix} 1 \\ z \\ \vdots \\ z^\psi \end{pmatrix}.$$

Then C has access to the following information:

1. The verification algorithms $\tilde{v}_1, \ldots, \tilde{v}_\omega$. We have, for $U_\ell \in C$,

$$\tilde{v}_\ell(x, z) = \left(1\ x \cdots\ x^{n-1}\right)\left(A_0 + v_{\ell,1}A_1 + \ldots v_{\ell,\omega}A_\omega\right)\begin{pmatrix}1\\ z\\ \vdots\\ z^\psi\end{pmatrix}.$$

   Noting that $\tilde{v}_\ell$ is a polynomial with terms of the form $(c_{ik})_\ell x^i z^k$ for $0 \leq i \leq n-1$ and $0 \leq k \leq \psi$, we have that C has access to $n(\psi+1)(\omega)$ equations $(c_{ik})_\ell$, where

$$(c_{ik})_\ell = a_{i0k} + \sum_{j=1}^{\omega} a_{ijk} v_{\ell,j}.$$

2. The signing algorithms $s_1, \ldots, s_\omega$. We have

$$s_\ell = \left(1\ U_1 \cdots\ U_1^{n-1}\right)\left(A_0 + y_1A_1 + \ldots y_\omega A_\omega\right)\begin{pmatrix}1\\ z\\ \vdots\\ z^\psi\end{pmatrix}.$$

   Noting that s_ℓ is a polynomial with terms of the form $(d_{jk})_\ell y_j z^k$ (with y_0 understood to mean 1), for $0 \leq j \leq \omega$ and $0 \leq k \leq \psi$, we have that C has access to $(\omega+1)(\psi+1)(\omega)$ equations $(d_{jk})_\ell$, where

$$(d_{jk})_\ell = \sum_{i=0}^{n-1} a_{ijk} U_\ell^i.$$

3. Up to ψ signatures $\sigma_{t,k'}$ from each user $U_t \notin C$, on messages $m_{t,k'}$ of his choice, where $1 \leq k' \leq \psi$, with the exception that $\mathcal{C}$ can only access a signature $\sigma_{n,k'}$ on a message $m_{n,k'} \neq m$ with signer U_s. Thus C has access to $n - \omega$ signatures of the form

$$F(U_t, y_1, \ldots, y_\omega, m_{t,k'}) = \left(1\ U_j \cdots\ U_j^{n-1}\right)\left(A_0 + y_1A_1 + \ldots y_\omega A_\omega\right)\begin{pmatrix}1\\ m_{t,k'}\\ \vdots\\ m_{t,k'}^\psi\end{pmatrix}.$$

   Note that $\sigma_{t,k}$ is a polynomial with terms of the form $(b_j)_{t,k'} y_j$ (with y_0 understood to mean 1). Then C has access to $(\omega+1)(\psi)(n-\omega)$ equations, where

$$(b_{jk'})_t = \sum_{i=0}^{n-1}\sum_{k=0}^{\psi} a_{ijk} U_t^i m_{t,k'}^k.$$

4. Up to ψ_V query results from the oracle $\mathsf{Vrfy}_\ell^{\mathcal{O}}$ for $U_\ell \notin C$. In the following, we will first consider the attack scenario without $\mathsf{Vrfy}^{\mathcal{O}}$ queries and then move to incorporate these queries into the analysis.

Now, these equations are not a linearly independent set, due to the relationships between users' signing and verification algorithms. More specifically, for any users U_ℓ and U_t, we have

$$s_\ell(v_t, z) = \tilde{v}_t(U_\ell, z). \tag{2}$$

and for a signature $\sigma_{t,k'}$ on the message $m_{t,k'}$ we have

$$\sigma_{t,k'}(v_\ell) = \tilde{v}_\ell(U_t, m_{t,k'}). \tag{3}$$

Equation (2) implies that for each $U_\ell \in \mathcal{C}$ and each $0 \leq k \leq \psi$, we have a set of ω relations among the $\omega + 1$ equations $\{(d_{jk})_\ell : 0 \leq j \leq \omega\}$. Equation(3) implies that for each $U_t \notin \mathcal{C}$, we have a set of ω relationship among the $\omega + 1$ equations $\{(b_{jk'})_t : 0 \leq j \leq \omega\}$.

We note the equations $\{(c_{ik})_\ell : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 1 \leq \ell \leq \omega\}$ form a linearly independent set, since the rank of $\{[1, v_1], \ldots, [1, v_\omega]\} \subset \mathbb{F}_q^{\omega+1}$ is ω.

We thus take the equations $\{(c_{ik})_\ell : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 1 \leq \ell \leq r\}$, $\{(d_{0k})_\ell : 0 \leq k \leq \psi, 1 \leq \ell \leq \omega\}$, and $\{(b_{0k'})_t : 1 \leq k' \leq \psi, \omega + 1 \leq t \leq n\}$. These equations do form a linearly independent set; we do not include the proof here. To summarize, we have $n - \omega$ free variables in the given linear system.

With the given information, C can consider the polynomials $F'(x, y_1, \ldots, y_\omega, z)$ consistent with the known information about F. If a given polynomial F' is consistent with the known information about F, we say F' satisfies property $(*)$. From above, we have that the total number of polynomials F' satisfying $(*)$ is $q^{n-\omega}$.

Case: $U_\zeta \notin C$

Suppose first $U_\nu \in C$. Then the goal of C is to produce a ζ-authentic signature. Given that the condition for success does not depend on the particular target verifier's verification key, v_ν, we can calculate the probability of success as

$$\frac{|\{F'(x, y_1, \ldots, y_\omega, z)) : F' \text{ satisfies } (*), F'(U_\zeta, y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega, z)\}|}{|\{F'(x, y_1, \ldots, y_\omega, z) : F' \text{ satisfies } (*)\}|}.$$

Using the same notation as before, if $F'(U_\zeta, y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega z)$, we have the additional equations $\{(d_{0k})_\zeta : 0 \leq k \leq \psi\}$, rendering the equations $\{(b_{0k'})_\zeta : 1 \leq k' \leq \psi\}$ redundant. We can show the resulting set is linearly independent, so we have one additional restriction on F'. Recalling that we chose F' from a space of size $q^{n-\omega}$ initially, we have $\frac{q^{n-\omega-1}}{q^{n-\omega}} = \frac{1}{q}$ as the coalition $\mathcal{C}$'s probability of success.

Suppose C also has access to the $\mathsf{Vrfy}^{\mathcal{O}}$ oracle. Note that if the query (m, σ) to $\mathsf{Vrfy}_\ell^{\mathcal{O}}$ results in *True* (for some $U_\ell \notin C$), then C has successfully determined U_ζ's signing algorithm, s_ζ. That is, we have $\tilde{v}_t(U_\zeta, m) = \sigma(v_{t,1}, \ldots, v_{t,\omega})$ for $U_t \in C$ and $\tilde{v}_\ell(U_\zeta, m) = \sigma(v_{\ell,1}, \ldots, v_{\ell,\omega})$. As in the proof of Lemma A.3, this yields $\sigma(y_1, \ldots, y_\omega) = s_\zeta(m)$, so (m, σ) is a ζ-authentic signature. In this case, C actually has $\psi + 1$ ζ-authentic signatures, i.e. $F' = s_\zeta$, so C can produce signatures from U_ζ at will. The probability of this happening, however, is the probability of C choosing the correct F', which, as we show below, is $\frac{1}{q-\psi'}$, where ψ' is the number of queries to $\mathsf{Vrfy}^{\mathcal{O}}$ with result $False$.

Now consider ψ_F queries to $\mathsf{Vrfy}^{\mathcal{O}}$ with result $False$, supposing each query is consistent with C's view of the function F. We observe that each negative query eliminates (at most) one potential signing algorithm for U_ζ.

Given that the condition for success does not depend on the particular target verifier's verification key, v_ν, we can calculate the probability of success as before, this time allowing for information gleaned from the ψ_V negative queries. We write $\bar{s_\zeta}^1, \ldots, \bar{s_\zeta}^{\psi_F}$ for these eliminated signing algorithms, and for readability, we write F'_ζ for $F'(U_\zeta, y_1, \ldots, y_\omega, z)$.

We first need to calculate $\#\{F'(x, y_1, \ldots, y_\omega, z) : F' \text{ satisfies } (*), F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}\}$, i.e., the number of possible functions F' consistent with C's view of F. Letting $\mathcal{F} = \{F'(x, y_1, \ldots, y_\omega, z) : F' \text{ satisfies } (*)\}$, we have

$$\begin{aligned}&\#\{F' \in \mathcal{F} : F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}\}\\ &= \#\{F' \in \mathcal{F}\} - \#\{F' \in \mathcal{F} : F' = \bar{s_\zeta}^1 \text{ or } F' = \bar{s_\zeta}^2 \text{ or } \cdots \text{ or } F' = \bar{s_\zeta}^{\psi_F}\}.\end{aligned}$$

We will assume the events $F' = \bar{s_\zeta}^1, \ldots, F' = \bar{s_\zeta}^{\psi_F}$ are disjoint, since if $\bar{s_\zeta}^i = \bar{s_\zeta}^j$ for some $1 \leq i, j \leq \psi_F$, this is equivalent to fewer verification oracle queries. Following the same reasoning as before, we have

$$\begin{aligned}&\#\{F' \in \mathcal{F} : F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}\}\\ &= \#\{F' \in \mathcal{F}\}| - |\{F' \in \mathcal{F} : F' = \bar{s_\zeta}^1\} - \#\{F' \in \mathcal{F} : F' = \bar{s_\zeta}^2\} - \cdots - \#\{F' \in \mathcal{F} : F' = \bar{s_\zeta}^{\psi_F}\}\\ &= q^{n-\omega} - \psi_F q^{n-\omega-1} = q^{n-\omega-1}(q - \psi_F).\end{aligned}$$

We calculate C's probability of success as:

$$\begin{aligned}&\frac{|\{F'(x, y_1, \ldots, y_\omega, z)) : F' \text{ satisfies } (*), F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}, F'_\zeta = s_\zeta, \}|}{|\{F'(x, y_1, \ldots, y_\omega, z) : F' \text{ satisfies } (*), F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}\}|}\\ &= \frac{|\{F'(x, y_1, \ldots, y_\omega, z)) : F' \text{ satisfies } (*), F'_\zeta = s_\zeta, \}|}{|\{F'(x, y_1, \ldots, y_\omega, z) : F' \text{ satisfies } (*), F' \neq \bar{s_\zeta}^1, F' \neq \bar{s_\zeta}^2, \ldots, F' \neq \bar{s_\zeta}^{\psi_F}\}|}\\ &= \frac{q^{n-\omega-1}}{q^{n-\omega-1}(q - \psi_F)} = \frac{1}{q - \psi_F}.\end{aligned}$$

Now suppose $U_\nu \notin C$. Ostensibly the goal of C is to produce a (ζ, ν)-acceptable signature. Note that in order for a signature pair (m, σ) with claimed signer U_ζ to pass U_ν's verification algorithm, (m, σ) must satisfy $\sigma(v_\nu) = \tilde{v}_\nu(U_\zeta, m)$.

In particular, if C constructs (m, σ) using a polynomial F' of the same form as F, we must have $F'(U_\zeta, v_\nu, m) = F(U_\zeta, v_\nu, m)$. The following lemma shows that if F' also satisfies $(*)$, this condition is equivalent to $F'(U_\zeta, y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega z)$.

Lemma A.3. *Let $m \neq m_{\zeta,k'}$ for any $1 \leq k' \leq \psi$. Suppose F' is a polynomial consistent with the verification algorithms of C, such that $F'(U_\zeta, v_\nu, m) = F(U_\zeta, v_\nu, m)$. Then we have $F'(U_\zeta, y_1, \ldots, y_\omega, m) = F(U_\zeta, y_1, \ldots, y_\omega, m)$; that is, σ is a ζ-authentic signature on m. In addition, if F' is also consistent with the sample signatures $m_{\zeta,k'}$ from U_ζ, then F' satisfies $F'(U_\zeta, y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega z)$. That is, F' is U_ζ's signing algorithm $s_\zeta(y_1, \ldots, y_\omega, z)$.*

Proof. From the verification algorithms from $\mathcal{C}$, we have that $F'(U_\zeta, v_\ell, m) = F(U_\zeta, v_\ell, m)$ for any $U_\ell \in \mathcal{C}$. That is, F' and F agree as polynomials in the y_i's on the $\omega+1$ points $v_1, \ldots, v_\omega, v_\nu$. Since the set of vectors $R = \{[1, v_1], \ldots, [1, v_\omega], [1, v_\nu]\}$ is linearly independent, we can, by Lemma A.2, choose a linearly independent subset $R' \in \{v_1, \ldots, v_\omega, v_\nu\}$ of size ω.

Write $R' = \{r_1, \ldots, r_\omega\}$ and let $r' = [r'_1, \ldots, r'_\omega] \in \{v_1, \ldots, v_\omega, v_\nu\} - R'$. Since R' is a basis for $\mathbb{F}_q^\omega$, we can write $r' = \sum_{j=1}^{\omega} k_j r_j$, where $k_i \in \mathbb{F}_q$. Set

$$H(y_1, \ldots, y_\omega) = (F - F')(x, y_1, \ldots, y_\omega, z)|_{x=U_\zeta, z=m} = h_0 + \sum_{i=1}^{\omega} h_i y_i.$$

Then in particular, we have $H(r') = H(r_1) = \cdots = H(r_\omega) = 0$.

We have

$$H(r') = \sum_{j=1}^{\omega} k_j (H(r_j)) \tag{4}$$

$$\iff h_0 + \sum_{i=1}^{\omega} h_i r'_i = \sum_{j=1}^{\omega} k_j \left(h_0 + \sum_{i=1}^{\omega} h_i r_{j,i} \right) \tag{5}$$

$$\iff h_0 + \sum_{i=1}^{\omega} h_i \left(\sum_{j=1}^{\omega} k_j r_{j,i} \right) = h_0 \left(\sum_{j=1}^{\omega} k_j \right) + \sum_{j=1}^{\omega} k_j \sum_{i=1}^{\omega} h_i r_{j,i} \tag{6}$$

$$\iff h_0 + \sum_{i=1}^{\omega} \sum_{j=1}^{\omega} h_i k_j r_{j,i} = h_0 \left(\sum_{j=1}^{\omega} k_j \right) + \sum_{j=1}^{\omega} \sum_{i=1}^{\omega} h_i k_j r_{j,i} \tag{7}$$

$$\iff h_0 = h_0 \left(\sum_{j=1}^{\omega} k_j \right). \tag{8}$$

Equation (8) implies either $h_0 = 0$ or $\sum_{j=1}^{\omega} k_j = 1$. That $\sum_{j=1}^{\omega} k_j \neq 1$, however, follows from the linear independence of R. Recalling $r' = \sum_{j=1}^{\omega} k_j r_j$, we calculate

$$\begin{aligned} \sum_{j=1}^{\omega} k_j [1, r_j] &= \sum_{j=1}^{\omega} k_j [1, r_{j,1}, \ldots, r_{j,\omega}] \\ &= \left[\sum_{j=1}^{\omega} k_j, r'_1, \ldots, r'_\omega \right]. \end{aligned}$$

That is, if $\sum_{j=1}^{\omega} k_j = 1$, then the vector $[1, r']$ can be written as a linear combination of the vectors of $R - [1, r']$, thereby contradicting the linear independence of the set. We conclude $h_0 = 0$.

Now, we have that H is a polynomial of the form $\sum_{i=1}^{\omega} h_i y_i$ with ω zeros on the ω linearly independent vectors of R'. By Lemma A.1, we conclude that H is the zero polynomial. That is, $F'(U_\zeta, y_1, \ldots, y_\omega, m) = F(U_\zeta, y_1, \ldots, y_\omega, m)$.

Recall that $F'(U_\zeta, y_1, \ldots, y_\omega, m_{\zeta,k'}) = F(U_\zeta, y_1, \ldots, y_\omega, m_{\zeta,k'})$, where $1 \leq k' \leq \psi$ and the messages $m_{\zeta,k'}$ are distinct. Since we also have $F'(U_\zeta, y_1, \ldots y_\omega, m) = F(U_\zeta, y_1, \ldots, y_\omega, m)$, we have a total of $\psi+1$ points at which F' and F agree as polynomials in z. Since F' and F are polynomials of degree ψ in z, this is sufficient to conclude $F'(U_\zeta, y_1, \ldots, y_\omega, z) = F(U_\zeta, y_1, \ldots, y_\omega, z)$, as desired. □

Lemma A.3 follows from the fact that any signature pair (m, σ) with claimed signer U_ζ that is consistent with $\omega + 1$ verification algorithms must be ζ-authentic. Thus, the set of known information $(*)$ does not help create a (ζ, ν)-fraudulent signature. For the case of creating a (ζ, ν)-fraudulent signature, the most powerful collusion C includes the signer U_ζ, which we consider next.

Case: $U_\zeta \in C$, $U_\nu \notin C$

Here C's goal is to produce a (ζ, ν)-fraudulent signature. Recalling that $v_\nu \in \mathbb{F}_q^\omega$ is chosen uniformly at random, we see that the signing algorithms of C and sample signatures from $U_\ell \notin C$ have no bearing on the probability distribution for the key v_ν.

Given that for any subset of size $\omega + 1$, the corresponding subset of size $\omega + 1$ formed from the new vectors $[1, v_1], \ldots, [1, v_n] \in \mathbb{F}_q^{\omega+1}$ must be a linearly independent set, however, we see that knowledge of the keys v_ℓ for $U_\ell \in C$ does affect the probability distribution for the key v_ν. In particular, C is aware that $[1, v_\nu] \neq \sum_{j=1}^{\omega} k_j [1, v_j]$ for any choice of $\{k_1, \ldots, k_\omega \in \mathbb{F}_q : \sum_{j=1}^{\omega} k_j = 1\}$. That is, given $v_1, \ldots, v_\omega$, there are $q^\omega - q^{\omega-1}$ choices for v_ν, any of which are equally likely. We write V for the set of possible vectors v_ν.

Now suppose we want to create a (ζ, ν)-fraudulent signature $\sigma'(y_1, \ldots, y_\omega)$ on a message m. Suppose $\sigma(y_1, \ldots, y_\omega) = b_0 + \sum_{j=1}^{\omega} b_j y_j$ is the ζ-authentic signature on m. Then writing $\sigma'(y_1, \ldots, y_\omega) = b'_0 + \sum_{j=1}^{\omega} b'_j y_j$, we need $\sigma(v_\nu) = \sigma'(v_\nu)$, but $(b_0, \ldots, b_\omega) \neq (b'_0, \ldots, b'_\omega)$.

In other words, C needs to find a nonzero vector $\beta = [b_0 - b'_0, \ldots, b_\omega - b'_\omega]$ satisfying $\beta \cdot [1, v_\nu] = 0$. The probability of success is then calculated as

$$\begin{aligned} \max_\beta \frac{|\{v_\nu \in V : \beta \cdot [1, v_\nu] = 0\}|}{|\{v_\nu \in V\}|} &\leq \max_\beta \frac{|\{v_\nu \in \mathbb{F}_q^\omega : \beta \cdot [1, v_\nu] = 0\}|}{|\{v_\nu \in V\}|} \\ &= \frac{q^{\omega-1}}{q^\omega - q^{\omega-1}} = \frac{1}{q-1}. \end{aligned}$$

We now consider $\mathsf{Vrfy}^{\mathcal{O}}$ queries. We observe that a positive $\mathsf{Vrfy}_\nu^{\mathcal{O}}$ query (m, σ) allows the coalition C to win the game Sig-forge$_{C,\Pi}(k)$, so we consider the probability of success given ψ_F negative $\mathsf{Vrfy}_\nu^{\mathcal{O}}$ queries.

We let V' be the set of possible vectors v_ν given the new knowledge gleaned from the ψ_F negative query vectors $\beta_1, \ldots \beta_{\psi_F}$. That is, $V' = \{v_\nu \in V : \beta_1 \cdot [1, v_\nu] \neq 0, \ldots, \beta_{\psi_F} \cdot [1, v_\nu] \neq 0\}$.

Now,

$$\begin{aligned} \#\{v_\nu \in V'\} &= \#\{v_\nu \in V\} - \#\{v_\nu \in V : \beta_1 \cdot [1, v_\nu] = 0 \text{ or } \cdots \text{ or } \beta_{\psi_F} \cdot [1, v_\nu] = 0\} \\ &\geq \#\{v_\nu \in V\} - \#\{v_\nu \in \mathbb{F}_q^\omega : \beta_1 \cdot [1, v_\nu] = 0 \text{ or } \cdots \text{ or } \beta_{\psi_F} \cdot [1, v_\nu] = 0\} \\ &\geq \#\{v_\nu \in V\} - \#\{v_\nu \in \mathbb{F}_q^\omega : \beta_1 \cdot [1, v_\nu] = 0\} - \cdots - \#\{v_\nu \in \mathbb{F}_q^\omega : \beta_{\psi_F} \cdot [1, v_\nu] = 0\} \\ &= (q^\omega - q^{\omega-1}) - \psi_F q^{\omega-1} = q^{\omega-1}(q - \psi_F - 1) \end{aligned}$$

The probability of success is then calculated as

$$\begin{aligned} \max_\beta \frac{|\{v_\nu \in V' : \beta \cdot [1, v_\nu] = 0\}|}{|\{v_\nu \in V'\}|} &\leq \max_\beta \frac{|\{v_\nu \in \mathbb{F}_q^\omega : \beta \cdot [1, v_\nu] = 0\}|}{|\{v_\nu \in V'\}|} \\ &\leq \frac{q^{\omega-1}}{q^{\omega-1}(q - \psi_F - 1)} = \frac{1}{q - \psi_F - 1}. \end{aligned}$$