Galois Field Commitment Scheme

Alexandre Pinto André Souto Armando Matos Luís Antunes
University of Porto, Portugal

November 13, 2006

Abstract

In [3] the authors give the first mathematical formalization of an unconditionally secure commitment scheme. Their construction has some similarities to one used to build authentication codes, so they raise the question whether there is some relation between commitment schemes and authentication schemes. They conjecture that authentication schemes with arbitration can be used, but they stress that the information flows are different.

In this paper, we show that there is indeed a relation between unconditionally secure commitment schemes and unconditionally secure authentication schemes, and that an unconditionally secure commitment scheme can be built from such an authentication scheme and an unconditionally secure cipher system. This parallel is then used to analyse a new attack against commitment schemes that is the counterpart of the impersonation attack in an authentication system.

To investigate the opposite direction, we start by defining an optimal commitment system and showing that this must be a resolvable design commitment scheme as proposed in the aforementioned paper. Then, a proof is given that the resolvable design commitment schemes are a composition of an authentication system and a cipher system and the conclusion follows that this is the case for all optimal commitment systems.

We prove that there is a commitment scheme based on Galois Fields that uses the One-Time Pad as the cipher system, which to our knowledge is new in the literature. The main technique in the proof is the construction of an appropriate design for any n, originating an authentication system that is perfectly secure against deception attacks of levels 0 and 1. The commitment scheme here proposed uses only very simple operations and can be very efficiently implemented both in hardware and software.

Finally, we give a brief look at the possibility of building commitment schemes from other primitives.

Keywords: Commitment, Authentication, Unconditional Security, Galois Field.

1 Introduction

Commitment schemes were introduced by Blum ([4]). It is not possible to build unconditionally secure commitment schemes with only two parties, but Rivest ([8]) proposed the first unconditionally secure non interactive commitment scheme with a trusted initializer.

In [3], the authors begin a mathematical formalization of such commitment schemes. They also prove some lower bounds on the binding probabilities and propose and analyse implementations of optimally secure systems. They leave as an open problem the existence of some relation between these schemes and authentication codes.

In this paper, this question is answered affirmatively. We show that an unconditionally secure commitment scheme with trusted initializer can be built from

- a composition of an unconditionally secure authentication code without secrecy, without splitting and with no arbitration

- and an unconditionally secure cypher system.

The authentication code prevents the committing party's cheating, while the cipher system prevents the receiving party from knowing too much before the right time. This relation elicits a new attack that can be executed in a commitment scheme and is the counterpart of the impersonation attack of an authentication system. We give two lower bounds for its probability of success, both combinatorial and information-theoretic.

After this, we define the notion of optimal commitment scheme and show that such a scheme is a resolvable design commitment scheme as proposed in [3]. In the following, it is shown that resolvable design commitment schemes can also be decomposed into a cipher system and an authentication code.

We then propose a new efficient commitment scheme. The affine plane commitment scheme (see [3]) requires the computation of values in an algebra modulo a prime. A computer can not handle these numbers as efficiently as it handles powers of 2, since in this case some arithmetic operations can be broken down to bit shifts and bitwise logical operations which have very fast hardware implementations. These are usually mirrored by equally efficient Assembly instructions and therefore fast software implementations. So, we address the question whether a commitment scheme can be built for an alphabet of symbols of size $|S| = 2^n$ for some integer n rather than $|S| = p$ for prime p. This has the advantage of allowing the One-Time Pad to be used as a cipher system which is unconditionally secure and very fast to implement. We show that this is possible for every n, by building an appropriate Transversal Design and using a result due to Stinson ([13]) to turn it into an unconditionally secure authentication system. Our commitment scheme follows from composition with the One-Time Pad cipher system.

The organization of the paper is as follows: Section 2 gives some definitions and notation that will be used in the sequel, and formal definitions of unconditionally secure cipher, authentication and commitment systems. In Section 3, we analyse a new attack against commitment schemes and show how these can be built from a cipher system and an authentication code. Section 4 is devoted to the study the resolvable design scheme proposed in [3]: we show that any optimal scheme is of this form and that these schemes can always be decomposed in a cipher system and an authentication system. In Section 5, a commitment scheme based on Galois Fields is presented that, to our knowledge, is new in the literature. In Section 6, some brief considerartions are given about whether cipher systems and authentication codes can be replaced by some other systems as building blocks, and Section 7 contains some final remarks and possible directions for future work.

2 Preliminaries

We denote alphabets by calligraphic type, e.g. $\mathcal{P}, \mathcal{C}$. Depending on context, these alphabets can be seen as subsets of $\mathbb{N}$ or of $\{0,1\}^*$. Elements of these alphabets are usually represented by lowercase letters. The size of a set is denoted by $|\cdot|$. Random variables over these sets are represented by uppercase versions of the name of the set, like P, C and so on. Greek letters are reserved for some probabilities and real parameters not greater than 1.

Sometimes, functions of two arguments are written as parameterized functions of one argument. For example, $f(k,s)$ is the same as $f_k(s)$. The function $H(\cdot)$ and its variants denote Shannon's Entropy function.

The participants in the several protocols bear the standard names of the literature: Alice and Bob are the legitimate participants, Eve is a passive eavesdropper, Oscar is a malicious opponent with

complete power over the channel between Alice and Bob, and Trent is a trusted initializer. Alice is always the sender and Bob the receiver. In the commitment scheme, both Alice and Bob can be malicious and try to break the protocol.

We now give formal definitions of the cryptographic constructions used in this paper.

2.1 Cryptographic Systems

Definition 2.1. A cipher system is a tuple denoted $CP(\mathcal{P}, \mathcal{C}, \mathcal{K}, f(k,p))$ where $\mathcal{P}$ is the alphabet of plain text messages, $\mathcal{C}$ is the alphabet of cipher text messages , $\mathcal{K}$ is the alphabet of secret keys and $f(k,p)$ is a family of $|\mathcal{K}|$ functions parameterized by $k \in \mathcal{K}$ where each function $f_k : \mathcal{P} \mapsto \mathcal{C}$ is injective and defined for all $p \in \mathcal{P}$. $\diamond$

A cipher system is unconditionally secure if the random variables $P, K, C = f(K,P)$ satisfy:

$$H(P) = H(P|C).$$

Definition 2.2. An authentication code without arbitration, without splitting and without secrecy is a tuple denoted $AC(\mathcal{S}, \mathcal{S} \times \mathcal{A}, \mathcal{K}, f(k,s), g(k, \langle s,a \rangle), \alpha, \beta)$ where $\mathcal{S}$ is the set of source states, $\mathcal{A}$ is the set of authenticators, $\mathcal{K}$ is the set of the secret keys, α is the maximum chance of success for an impersonation attack, β is the maximum chance of success for a substitution attack, $f(k,s)$ is a family of $|\mathcal{K}|$ functions parameterized by $k \in \mathcal{K}$ where each function $f_k : \mathcal{S} \mapsto \mathcal{A}$ is injective and defined for all $s \in \mathcal{S}$, $g(k, \langle s,a \rangle)$ is a family of $|\mathcal{K}|$ functions where each function $g_k : \mathcal{S} \times \mathcal{A} \mapsto \{0,1\}$ is defined for all $\langle s,a \rangle \in \mathcal{S} \times \mathcal{A}$. $\diamond$

We give only the details needed for authentication codes and refer the reader to [9], [10] and [11] for more information.

Authentication codes without secrecy as defined above allow a party to send a message composed of a source value s and an authenticator a such that an attacker has at most a probability α of forging a new message or a probability β of altering a known valid message such that the receiver will accept these forgeries as valid. If the attacker sees i valid messages before sending his forgery, this is called a deception attack of level i. The most basic attacks are the impersonation attack ($i = 0$) and the substitution attack ($i = 1$) and are the only ones considered in this paper. The probability of success for an attack of level i is denoted P_{d_i}.

The participants of the scheme share a secret key that allows the sending party to compute the right authenticator for a source value, by computing $a = f(k,s)$, and the receiving party to decide if a message is a forgery or not, by evaluating $g(k, \langle s,a \rangle)$.

There is always some positive probability of success for any attack. We list some bounds from the literature: $\log P_{d0} \geq H(K|M) - H(K)$ ([9]), $\log P_{d1} \geq -H(K|M)$ ([6]), $P_{d_0} \geq \frac{|\mathcal{S}|}{|\mathcal{M}|}$ and $P_{d_1} \geq \frac{|\mathcal{S}|-1}{|\mathcal{M}|-1}$ ([12], [13]). An authentication code is unconditionally secure if the maximum probabilities of success meet these bounds.

Definition 2.3. A commitment scheme is a tuple denoted $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ where $\mathcal{X}$ is the source states alphabet, $\mathcal{Y}$ is the coded states alphabet, $\mathcal{K}$ is the alphabet of the committer's keys, $\mathcal{V}$ is the alphabet of the verifier's tags, $f(k,x)$ is a family of $|\mathcal{K}|$ functions parameterized by $k \in \mathcal{K}$ where each function $f_k : \mathcal{X} \mapsto \mathcal{Y}$ is injective and defined for all $x \in \mathcal{X}$ and $g(v,k)$ is a family of $|\mathcal{V}|$ functions parameterized by $v \in \mathcal{V}$ where each function $g_v : \mathcal{K} \mapsto \{0,1\}$ is defined for all $k \in \mathcal{K}$. α and β are the maximum chances of success for the two kinds of attack available to the committer: α represents an attack where the committer ignores her key, whereas β represents an attack where the committer uses her key to guess the verifier's tag. These attacks are described in Section 3.1. $\diamond$

We give a brief explanation of commitment schemes with a trusted initializer. These schemes allow a sender to commit to a value and send that commitment to a receiver such that the value she committed to remains hidden from this. In a second step, the sender reveals her commitment and the receiver may verify that the sender is not fooling him. The third participant is required only to give the other two some information that enables them to carry out the protocol. This third participant is completely honest and trusted by the other two.

The initializer gives the sender a secret key k with which she can hide her commitment; he also gives the receiver a verifier tag v with which he can verify the key that the sender will reveal later. To commit to a value x, she computes $y = f(k, x)$ and sends the value to the receiver. To open the commitment, she sends him k' and since every f_k is injective, knowing k' the receiver can compute the inverse $x' = f_k^{-1}(x)$. To verify that $k' = k$ and therefore $x' = x$, the receiver computes his verifying function $g(v, k')$ and accepts or rejects accordingly.

A commitment scheme is required to be perfectly concealing, i.e., the receiver can guess the value committed to only with a probability equal to a uniform random guess. This is called the Concealing Property. On the other hand, the sender's commitment must effectively bind her, which means she can not open to the receiver a value different from her commitment. As shown in [3], a commitment system can not be completely binding, and so we say a system is $(1 - \epsilon)-$ binding if the probability of the sender deceiving the receiver is at most ϵ.

2.2 Combinatorial Designs

Design theory is a large body of research dedicated to statistical constructions known as designs. We give only the results and definitions we need in this paper and refer the reader to some textbook in design theory.

Definition 2.4. A $t-(v, k, \lambda)$ design is a pair $(\mathcal{D}, \mathcal{S})$ where $t \leq k < v$, $\lambda > 0$, $\mathcal{S}$ is a set of v distinct elements, called points, and $\mathcal{D}$ is a collection of subsets of $\mathcal{S}$ each with exactly k elements, called blocks. Besides, every point occurs in exactly r blocks and every subset of $\mathcal{S}$ with exactly t points occurs in exactly λ blocks. Henceforth, let $b = |\mathcal{D}|$.

These constructions are called $t-$designs. When $t = 2$, they are usually called Balanced Incomplete Block Designs (BIBD). $\diamond$

Definition 2.5. A design is said to be resolvable if its blocks can be partitioned into sets $\mathcal{P}_i$ called parallel classes, each with exactly v/k elements, such that the blocks in each parallel class form a partition of $\mathcal{S}$.

A resolvable $1-(v, k, \lambda)$ design is called affine if for any two blocks B_1, B_2 belonging to different parallel classes, it happens that $|B_1 \cap B_2|$ is equal to k^2/v. $\diamond$

Theorem 2.1. *If $(\mathcal{D}, \mathcal{S})$ is a $t-(v, k, \lambda)$ design, then $b = \dfrac{\lambda\binom{v}{t}}{\binom{k}{t}}$. Furthermore, each point occurs in exactly $r = \dfrac{\lambda\binom{v-1}{t-1}}{\binom{k-1}{t-1}}$.*

Definition 2.6. A transversal design $TD(k, n, \lambda)$ is a pair $(\mathcal{D}, \mathcal{S})$ such that $|\mathcal{S}| = k \cdot n$, the points in $\mathcal{S}$ can be divided into exactly k groups of n elements each, there are $\lambda \cdot n^2$ blocks, each of them containing at most one point from each group, and any pair of points from distinct groups occurs in exactly λ blocks. $\diamond$

It is easy to see from the definitions that a transversal design is not a $2-$design because two points from the same group are never contained in any block.

3 Analysis of Commitment Schemes

This section presents an analysis of the possible attacks against a commitment scheme and shows how to build such schemes from a cipher and an authentication scheme.

3.1 Security

In a commitment scheme, both participants can launch attacks.

Bob Attacks The security of a commitment scheme can be measured by the probability that Alice has of cheating Bob while Bob can not cheat Alice with more than a priori probability. Bob's chances at guessing each x should not be altered by his knowledge of v and y, i.e., for all triples (x, y, v), $\Pr[X = x|Y = y, V = v] = \Pr[X = x]$. This can be summarized using Shannon's entropy with $H(X) = H(X|Y, V)$.

Alice Attacks Let Alice commit to a value x and send $y = f(k, x)$ to Bob where k is her secret key. Alice cheats Bob if she can reveal a $k' \neq k$ such that $f_{k'}^{-1}(y) = x'$ with $x' \neq x$, and Bob accepts k' as valid, i.e., $g_v(k') = 1$.

It is proved in paper [3] that a Commitment Scheme can not be invulnerable against all of Alice's attacks. To see this, define the set of tags that validate a key k as $\mathcal{V}_k = \{v \in \mathcal{V} : \Pr[V = v|K = k] > 0\}$. Alice attacks like this: she computes the set $\mathcal{V}_k$ of all the tags that Bob may have. She then picks the tag $v_0 \in \mathcal{V}_k$ that maximizes $\Pr[V = v|K = k]$. Let $\alpha = \Pr[V = v_0|K = k]$. By an averaging argument, $\alpha \geq 1/|\mathcal{V}_k|$. Now, Alice picks two values $x \neq x'$ and computes $y = f(k, x)$. But by the concealing property, there is a key k' such that $f(k', x') = y$ and $g(v_0, k') = 1$ which allows Alice to cheat successfully if Bob's tag is v_0. The probability of this attack is the probability that Bob is holding the tag chosen by Alice, that is, α. So, Alice can always cheat with probability at least $1/|\mathcal{V}_k|$. It is shown in the same paper that the average probability of this attack is at least $2^{-H(V|K)}$ and therefore there's at least one instance with probability of success at least that big.

There is yet another attack that Alice can perform. The attack described above is the counterpart to a substitution attack in an authentication system. The following is the counterpart of an impersonation attack. These relations are a consequence of the construction of commitment schemes from authentication codes.

In the previous attack, Alice makes best use possible of her private information, but she can also launch an attack ignoring it altogether. To do this, Alice simply computes for each key the probability that Bob accepts it, i.e., for a fixed key k she finds

$$\gamma(k) = \sum_{v \in \mathcal{V}_k} \Pr[V = v] \tag{1}$$

She then picks the key that maximizes the above sum and reveals it to Bob in the revealing step.

We give two combinatorial lower bounds for this attack when the distribution of the keys and tags is uniform.

Theorem 3.1. *There is some $k \in \mathcal{K}$ with probability of success $\gamma(k) \geq \frac{E(|K_v|)}{|\mathcal{K}|}$, where $E(|K_v|)$ signifies the average number of keys that each tag validates.*

Proof. Consider $\gamma(k)$ as defined above. Its average value is

$$1/|\mathcal{K}| \sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[V=v] =$$
$$1/|\mathcal{K}| \sum_{v\in\mathcal{V}} |\mathcal{K}_v| \Pr[V=v] =$$
$$\frac{E(|K_v|)}{|\mathcal{K}|}$$

Then, by an averaging argument, there is some k which has

$$\gamma(k) \geq \frac{E(|K_v|)}{|\mathcal{K}|}$$

□

Corollary 3.1. *There is some $k \in \mathcal{K}$ with probability of success $\gamma(k) \geq \frac{E(|\mathcal{V}_k|)}{|\mathcal{V}|}$, where $E(|\mathcal{V}_k|)$ signifies the average number of tags that validate each key.*

Proof. It suffices to note that $\sum_{k\in\mathcal{K}} |\mathcal{V}_k| = \sum_{v\in\mathcal{V}} |\mathcal{K}_v|$. □

We can show an analog result with Shannon's entropy

Theorem 3.2. *There is some $k \in \mathcal{K}$ with probability of success*

$$\gamma(k) \geq 2^{-I(K;V)}$$

Proof. By definition of mutual information (see [5])

$$-I(K;V) = \sum_{k\in\mathcal{K},v\in\mathcal{V}} \Pr[K=k,V=v] \log \frac{\Pr[K=k]\Pr[V=v]}{\Pr[K=k,V=v]}.$$

For each $k \in \mathcal{K}$, $\Pr[K=k,V=v] = 0$ for every $v \notin \mathcal{V}_k$. Thus, the above can be written

$$\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k,V=v] \log \frac{\Pr[K=k]\Pr[V=v]}{\Pr[K=k,V=v]} \tag{2}$$

The log sum inequality (see [5]) states that

$$\sum_{i=1}^{n} a_i \log \frac{b_i}{a_i} \leq \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} b_i}{\sum_{i=1}^{n} a_i}. \tag{3}$$

Applying this to (2),

$$\begin{aligned}
-I(V;K) &= \sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k,V=v] \log \frac{\Pr[K=k]\Pr[V=v]}{\Pr[K=k,V=v]} \\
&\leq \left(\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k,V=v]\right) \log \frac{\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k]\Pr[V=v]}{\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k,V=v]} \\
&= 1 \cdot \log \sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{V}_k} \Pr[K=k]\Pr[V=v] \\
&= \log \sum_{k\in\mathcal{K}} \Pr[K=k]\gamma(k) \\
&= \log E(\gamma(K))
\end{aligned}$$

where $E(\gamma(K))$ is the average value of the success probability for each k. By an averaging argument, there is at least one k that has probability greater or equal to the average value. For this k:

$$\gamma(k) \geq E(\gamma(K)) \geq 2^{-I(K;V)}$$

□

In conclusion, there are two attacks available to the committer. The log of the probability of these attacks is lower bounded by $-H(K|V)$ and by $-I(K;V) = -H(K) + H(K|V)$.

A Commitment Scheme is said to be unconditionally secure if it is perfectly concealing and the maximum probabilities of success for these two attacks are equal and meet the lower bounds. This implies $H(K) = 2H(K|V)$.

3.2 Construction of Commitment Schemes

This section presents a proof that an unconditionally secure commitment scheme can be built using an unconditionally secure cipher system and an unconditionally secure authentication system without secrecy as building blocks.

Since we are going to compose a commitment scheme with two different systems, the users of the former will play different roles of these systems at different steps, so we refer to these roles by writing the abbreviation of the system followed by the role played, all within square brackets.

In both a cipher scheme and an authentication scheme, the system only needs to defend against one attacker. But in the commitment scheme, there are two different kinds of attacker, and each has different possibilities at his disposal. So, the commitment scheme has to employ two different kinds of defense.

One attack is against secrecy: Bob must not learn the secret value Alice committed to before the right time, so Alice sends it enciphered. The second attack is against authentication: Alice must not send a fake opening key, so she must send it through an authentication scheme. In the first step, Alice uses a cipher system without receiver. She merely sends Bob an encrypted message, but he must not be able to open it. Essentially, Bob takes the role of [CP.Eve]. After Bob receives a key in the revealing step, he takes the role of [CP.Bob] and opens the cipher text learning Alice's commitment.

In the second step, Alice sends Bob the key to open the encrypted message he has, but Bob needs to be sure it is the right key, this is, the one that was distributed to her in the initial phase. Essentially, Alice acts as a relay between Trent, the initializer of the commitment scheme, and Bob. Since she reads what the initializer sends and has a choice of relaying that message or changing it for another one altogether, she has complete control over the channel. She is indeed the attacker, now. In this phase, Trent plays the role [AC.Alice], Alice plays the role [AC.Oscar] and Bob plays the role [AC.Bob]. We summarize the above in Table 1.

User	**Committing Step**	**Revealing Step**
Alice	[CP.Alice]	[AC.Oscar]
Bob	[CP.Eve]	[CP.Bob] / [AC.Bob]
Trent	...	[AC.Alice]

Table 1: Roles Played

Theorem 3.3. *Given an unconditionally secure cipher system $CP(\mathcal{P}, \mathcal{C}, \mathcal{K}, f(k,p))$ and an authentication system without secrecy $AC(\mathcal{S}, \mathcal{S}\times\mathcal{A}, \mathcal{E}, h(e,s), g(e,\langle s,a\rangle), \alpha, \beta)$ with $\mathcal{S} = \mathcal{K}$, there is a commitment scheme with initializer (per Rivest's model) $CM(\mathcal{P}, \mathcal{C}, \mathcal{S}\times A, \mathcal{E}, f(s,p), g(e,\langle s,a\rangle), \alpha, \beta)$.*

Proof. The several components of the commitment scheme are obtained from the cipher and the authentication system as shown in Table 2. Similar names are distinguished by prefixing them with the abbreviation of the system they come from.

Cipher	**Commit**	**Auth**
$\mathcal{P}$	$\mathcal{X}$	...
$\mathcal{C}$	$\mathcal{Y}$	...
$\mathcal{K}$	...	$\mathcal{S}$
...	$\mathcal{K}$	$\mathcal{S} \times \mathcal{A}$
...	$\mathcal{V}$	$\mathcal{E}$
$f(k,p)$	$f(k,x)$	...
...	$g(k,v)$	$g(k,\langle s,a\rangle)$

Table 2: Equivalences Between Systems

Because we have used letters given in the initial definitions, there are two different alphabets labeled $\mathcal{K}$. They are not to be confused. For each $k \in CP.\mathcal{K}$, there are $|\mathcal{A}|$ different $k' \in CM.\mathcal{K}$, all with the same behaviour in the cipher system. Considering the analysis in [3], these $|\mathcal{A}|$ keys form a parallel class of keys in the combinatorial design used as basis for the resolvable design commitment scheme.

The protocol is as follows:

1. **Initialization:** Trent chooses randomly a value $s \in \mathcal{S}$ and then an authenticator a. It may happen that for a given s the authenticators are not distributed uniformly, but it is crucial that the several s are. Then, he randomly chooses one key e that authenticates $\langle s,a\rangle$ and sends $\langle s,a\rangle$ to Alice and e to Bob. For the value held by Bob, Alice's message is a valid authentication of source value s.

2. Alice extracts s from the message Trent sent her. She can do this because the authentication system has no secrecy. Since by definition $\mathcal{S} = \mathcal{K}$, Alice now has a uniformly randomly chosen key for the cryptosystem: $s = k \in CP.\mathcal{K}$.

3. **Committing Step:** Alice chooses a value x to commit to. She enciphers it with $y = f(k,x)$ and sends y to Bob.

4. **Revealing Step:** Alice sends Bob a possibly false key $\langle s',a'\rangle$. Bob checks its validity with $g_v(\langle s',a'\rangle)$ and if he accepts it, he extracts s' and computes $x' = f_{s'}^{-1}(y)$, opening the commitment.

This construction yields a commitment scheme that follows Rivest's model, as is shown next. The crucial point of this construction is that the source value that Trent wants to send Bob in the revelation step is the actual key that the latter must use to open Alice's commitment. The figures in Appendix B can help to understand this.

It is easy to verify that the families of functions $f(k,p)$ and and $g(k,\langle s,a\rangle)$ satisfy the formal requirements of the commitment scheme. Now we check the concealing and binding properties.

Concealing Let x_0 be the value Alice committed to. For Bob to find it, he must be able to invert the cipher text he received. Denote by k_0 the key held by Alice and by $y_0 = f(k_0,x_0)$ the value Bob received. This k_0 will be sent to Bob later by an authentication scheme where it plays the role of a source value. But for now, Bob does not know k_0. However, he does have a validation tag and this

plays the role of a key in the authentication scheme, so we have to show that this does not improve his chances of guessing x_0. Let v_0 be Bob's tag. Then, because the authentication system does not have splitting, for each possible source value k and each key v_0, $h(v_0, k)$ produces exactly one pair (k, a), where a is some authenticator.

But this means that Bob's tag can validate exactly one message for each possible key of the cipher system and so the set of possible messages for Bob has the same entropy as the set of keys for the cipher system. Bob is exactly in the same position he would be if he did not have his tag, and without Alice's key, Bob must break an unconditionally secure cipher system to find her commitment. Then, this commitment scheme satisfies the concealing property.

Binding Let x_0 be the value Alice committed to and k_0 be the key for the cipher system that she holds. In order to reveal a value $x' \neq x_0$, Alice needs to make Bob accept a key $k' \neq k_0$. But since $k = s$, this is the same as succeeding in an attack against the system by sending some $\langle s', a' \rangle \neq \langle s, a \rangle$ that makes Bob accept a fraudulent source value $s' \neq s$ which would make Bob open a $x' \neq x$.

Alice knows a valid coded message, so she could launch a substitution attack, but if the probability of an impersonation were higher, she could always act as if she had never known the valid message and simply try to forge one. This attack has been described in Section 3.1. It can now be seen that it corresponds exactly to the impersonation attack in an authentication system. Therefore, the probabilities of success for the two described attacks against the commitment scheme are the probabilities of success for an impersonation and a substitution attack against the authentication code. Thus, this commitment is $(1 - \max(\alpha, \beta))-$binding. □

Corollary 3.2. *Given an unconditionally secure cipher system and an unconditionally secure authentication system without secrecy there is an unconditionally secure commitment scheme with initializer (per Rivest's model).*

In the appendix, we show how the different flows of information in the three systems are related. This is not necessary to understand the proof and is merely presented for completeness' sake.

4 Optimal Commitment Schemes

In [3], the authors propose a general commitment scheme which they called Resolvable Design Commitment Scheme. In this section, we define optimal commitment schemes and show that they are resolvable design affine commitment schemes. Then, we close the circle showing that all resolvable design commitment schemes can be viewed as the composition of a cipher system and an authentication system. For simplification, in what follows, consider that the source values, keys and verification tags are distributed uniformly, since this maximizes uncertainty and therefore security.

The proof that optimal commitment schemes are affine resolvable is long and includes several lemmas, so we list only the results and leave the proofs for the appendix.

Definition 4.1. A commitment scheme $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k, x), g(v, k), \alpha, \beta)$ is optimal if it is unconditionally secure, $|\mathcal{X}| = |\mathcal{Y}|$ and has the minimum number of keys for a fixed number of source states and the desired security level. Besides, the probability of Alice's cheating should be equal to the probability of Bob's cheating. ⋄

The Lemmas 4.1 to 4.4 give some properties that an optimal commitment system must have. Lemma 4.3 excludes BIBDs as the possible minimal system, and this is necessary because such systems are not resolvable. This means that there can be pairs of blocks with empty intersection and these are counted in Lemma 4.4.

After these lemmas, we're ready to give the two main theorems: that an optimal commitment system must be affine resolvable and that a resolvable commitment scheme can be decomposed into a cipher system and an authentication system.

Lemma 4.1. *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then* $\alpha = \beta$ *and* $|\mathcal{V}| = (1/\alpha)^2$.

Lemma 4.2. *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then* $|\mathcal{X}|^2 = |\mathcal{K}| = |\mathcal{V}|$.

Lemma 4.3. *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then its incidence matrix can not be a BIBD.*

Lemma 4.4. *Let* $p = |\mathcal{X}|$. *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then the sum of distinct pairs of keys that don't have any tag in common is* $p^2 \cdot (p-1)/2$.

Theorem 4.1. *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then it is an affine resolvable commitment scheme, and all keys in each parallel class encrypt each value* $x \in \mathcal{X}$ *to the same value* $y \in \mathcal{Y}$. *That is, the function* $y = f(k,x)$ *depends only on the index of the parallel class containing* k, *and not on* k *itself.*

Theorem 4.2. *A resolvable design commitment scheme obtained from a resolvable* $1-(v,k,\lambda)$ *design* $(\mathcal{D}, \mathcal{S})$ *is a composition of a perfectly secure cipher system and an authentication code with* $P_{d0} = k/v$ *and* $P_{d1} = \frac{\max |B_1 \cap B_2|}{k}$ *for all* $B_1, B_2 \in \mathcal{D}, B_1 \neq B_2$.

5 Galois Field Commitment Scheme

As noted in [3], affine resolvable 2−designs are optimal among the resolvable designs in terms of binding probabilities, however not many classes of such designs are known to exist. Notwithstanding this, there are other kinds of designs that can achieve the same goals, namely Transversal Designs.

This section addresses this question by showing how to construct a resolvable Transversal Design $TD(2^n, 1, 2^n)$, for any n, that is also a $1-(2^{2n}, 2^n, 1)$ affine resolvable design. From such a design, we then build an unconditionally secure authentication code and an unconditionally secure commitment scheme.

Theorem 5.1. *For any positive integer* n, *it is possible to construct a Transversal Design* $TD(2^n, 1, 2^n)$.

Proof. It is well known from Algebra that finite fields exist if and only if the order of the field is prime or the power of a prime (see [2]) and that for any such order there is a unique finite field up to an isomorphism. These finite fields are usually denoted $GF(p^n)$, for Galois Field of order p^n. So, for any n there is a Galois Field of order 2^n. For composite orders p^n, the elements of the field are considered to be polynomials and the operations of the field are addition and multiplication of polynomials modulo the prime p. Addition is denoted by $\oplus$ and multiplication by $\odot$.

It is known that for any finite field $GF(p^n)$ a primitive polynomial of degree n and coefficients modulo p exists ([2]). Fix such a polynomial. Now, build a table with 2^{2n} rows and 2^n columns. Divide the rows in 2^n groups of 2^n elements each, and number each group from 0 to $2^n - 1$ and each row in the group likewise. Thus, each row is numbered by a pair (a, b) where $a, b \in \mathbb{Z}_{2^n}$. Analogously, associate a number $x \in Z_{2^n}$ to each column. Now, in cell $((a,b), x)$, place the value $a \odot x \oplus b$. The result is a number in $\mathbb{Z}_{2^n}$.

We now show this table represents a $TD(2^n, 1, 2^n)$ transversal design. Consider the set of points $\mathcal{V} = \{0, 1, \ldots, 2^{2n} - 1\}$ and divide them in 2^n groups of 2^n points. Each row represents a block with

2^n points, one from each group. For row j, the i^{th} value represents the index of the point in the i^{th} group that belongs to the j^{th} block. By construction, each block has one point in each group. Finally, each two points from distinct groups can occur in only one block. To see this, let (x_0, y_0) and (x_1, y_1) be two points from distinct groups, where $x_0 \neq x_1$. For both points to be in the same block, there must be a pair (a, b) such that $a \odot x_0 \oplus b = y_0$ and $a \odot x_1 \oplus b = y_1$. Then,

$$a \odot x_0 - y_0 = a \odot x_1 - y_1 \Leftrightarrow$$
$$a \odot (x_0 - x_1) = y_0 - y_1$$

Since $(x_0 - x_1)$ and $(y_0 - y_1)$ are defined and $(x_0 - x_1) \neq 0$, then a is completely determined, and so is b. That means there is only one pair satisfying both equations, which means both points can belong to only one block. This concludes the proof. □

This design originates an authentication code $AC(2^n, 2^{2n}, 2^{2n}, 1/2^n, 1/2^n)$, as proved in [13].

With such an unconditionally secure authentication scheme, and using the One-Time Pad as a perfect cipher system, we can build an unconditionally secure Commitment Scheme as outlined in section 3.2.

Let $\mathcal{S} = \{0, 1\}^n$. The protocol is as follows:

1. **Initialization:** In the first step, Trent chooses randomly a pair $(a, b) \in \mathcal{S} \times \mathcal{S}$. He chooses randomly a number $x_1 \in \mathcal{S}$ and computes $y_1 = a \odot x_1 \oplus b$. He sends the pair (x_1, y_1) to Bob and the pair (a, b) to Alice.

2. **Committing Step:** Alice chooses a value $x_0 \in \mathcal{S}$ she wants to commit to, and using her secret key she computes an enciphered value $y_0 = x_0 \oplus a$. She sends y_0 to Bob.

3. **Revealing Step:** Alice sends Bob a message (a', b'). Bob checks the validity of the message by computing $a' \odot x_1 \oplus b'$. He accepts if it equals y_1 and rejects otherwise. If he accepts, he computes $x'_0 = y_0 \oplus a'$ and accepts x'_0 as Alice's commitment.

The resulting commitment scheme is very similar to the affine plane commitment scheme, but has some advantages worth noting. First of all, it uses the One-Time Pad as cipher system, which not only is well known to be secure but also is very fast to implement both in software and hardware. Besides, it allows the use of a complete alphabet of strings of size n, whereas in the affine plane with order p, the alphabet of allowed values does not coincide with any alphabet of all strings of a given size. In general, the latter systems will be less efficiently implemented in hardware and software because the basic instructions are more oriented to a fixed size of bits than the corresponding arithmetic value.

Finally, we address the matter of calculating addition and multiplication in a Galois Field. Each element of $GF(2^n)$ is a binary string of size n, where bit i corresponds to the coefficient of term x^i in a polynomial of degree strictly less than n. Addition is performed by adding the polynomial coefficients degree by degree, which corresponds easily to an exclusive-or between both strings. Thus, this operation can be performed extremely fast both in hardware and software, especially if n corresponds to the size of the word of the microprocessor used.

Multiplication can be computed by shifting one of the strings to the left an appropriate number of positions for each bit 1 in the other string, XORing the several displaced versions together and computing the remainder of the division by the primitive polynomial. This sounds complicated, but can all be implemented with shift and XOR instructions, and both kinds are quickly implemented in hardware and low-level software (Assembly Code). A whole analysis of a possible implementation is given in [7].

6 Brief Note on extending the system

It is worth considering the case whether cipher systems can be replaced by some other kind of cryptographic system in the construction proposed in this paper. After all, more cryptographic systems can hide a source value from someone who lacks a key, for example, authentication systems with secrecy or secret sharing schemes. But these constructions are more complicated than a cipher system because they do something else than simply masking a value.

On the other hand, as was evident in the affine plane example, some functions that at first do not seem to be cipher systems may be perfect for the role. Any family of n different permutations over n different values will be suitable for the system. But this is, after all, the description of a cipher system with a key as long as the source value. It suffices to choose the key and the source value independently and uniformly for us to get an unconditionally secure cipher system.

7 Conclusion and Further Work

This paper continues the work began in [3] in the analysis of unconditionally secure commitment schemes. It gives formal characterizations of cipher schemes, authentication schemes without secrecy, splitting or arbitration and perfectly concealing commitment schemes with a trusted initializer, as proposed by Rivest. Then, we showed how to build an unconditionally secure commitment scheme using an unconditionally secure cipher system and an unconditionally secure authentication code. Based on this construction, we showed an attack against commitment schemes analog to the impersonation attack of authentication codes and gave two lower bounds for its success probability, one combinatorial and one information theoretic. Then we showed that the resolvable design schemes proposed in that paper can be built according to our composition and that any optimal commitment system must be a system of this kind. We then considered whether it would be possible to build commitment schemes using the One-Time Pad as the cipher system. This requires the source alphabet to have 2^n elements. A positive answer was given to this question by showing how to build an adequate transversal design and then an authentication code with a source alphabet of size 2^n that is unconditionally secure against impersonation and substitution attacks. With the composition given before, this implies the existence of such commitment schemes which we called Galois Field Commitment Schemes.

This answers affirmatively the question raised in [3] about the relation of commitment schemes and authentication schemes, although with a solution different from the one suggested in that paper. Also this shows that a cipher system is needed too. Accordingly, the theory of authentication codes without secrecy can be used in the analysis of commitment schemes. Further work can be done to understand if other kinds of authentication codes can also be used to develop different kinds of commitment schemes and if there are viable alternatives to the cipher system.

Appendix A: Proof of Theorems 4.1 and 4.2

This section presents the proofs omitted earlier in Section 4.

Lemma 4.1 *If a commitment scheme* $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ *is optimal, then* $\alpha = \beta$ *and* $|\mathcal{V}| = (1/\alpha)^2$.

Proof. The probability of success α is the maximum probability, among all $k \in \mathcal{K}$, that it is validated by a verifier's tag. This can be written:

$$\alpha = \max_{k \in \mathcal{K}} |\mathcal{V}_k|/|\mathcal{V}| \geq E(|\mathcal{V}_k|)/|\mathcal{V}|$$

Thus α is minimum when the average $|\mathcal{V}_k|$ is equal to the maximum, which means, when $|\mathcal{V}_k|$ is constant for all k. In the same, and using the bound in Theorem 3.1, it happens that $|\mathcal{K}_v|$ must also be constant. In the remainder of this proof, assume these values to be constant for all k and v.

The probability β is the maximum probability that for a valid key k Alice can find a key $k' \neq k$ such that Bob accepts k' as valid. Let $\mu = \max_{k \neq k' \in \mathcal{K}} |\mathcal{V}_k \cap \mathcal{V}_{k'}|$. Then β can be written

$$\beta = \frac{\mu}{|\mathcal{V}_k|}$$

Clearly, β is minimum when $\mu = 1$ since μ can not 0. Otherwise, each tag would verify exactly one key and so Bob would be able to cheat Alice with absolute certainty.

We show that α must be equal to β. Let $\gamma = \max(\alpha, \beta)$. If the system is unconditionally secure, γ must be as low as possible. Suppose there is a relation of the tags and keys such that $\gamma_0 = \alpha_0 = \beta_0$. Now suppose we could find another relation that implied $\gamma_1 = \max(\alpha_1, \beta_1) < \gamma_0$. Assume w.l.o.g that $\alpha_1 > \beta_1$. Denote by $|\mathcal{V}_k^0|$ the value $|\mathcal{V}_k|$ in the first relation and by $|\mathcal{V}_k^1|$ the respective value in the second relation. Then,

$$\begin{aligned}
&\alpha_0 = \gamma_0 > \gamma_1 = \alpha_1 \Rightarrow \\
&|\mathcal{V}_k^1| < |\mathcal{V}_k^0| \Leftrightarrow 1/|\mathcal{V}_k^1| > 1/|\mathcal{V}_k^0| \Rightarrow \\
&\beta_1 > \beta_0 > \alpha_1
\end{aligned}$$

Thus we have a contradiction, and so the minimum value is achieved when $\alpha = \beta$. Now this implies that $1/|\mathcal{V}_k| = |\mathcal{V}_k|/|\mathcal{V}|$ whence we get that $|\mathcal{V}| = |\mathcal{V}_k|^2 = (1/\alpha)^2$. □

Lemma 4.2 If a commitment scheme $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ is optimal, then $|\mathcal{X}|^2 = |\mathcal{K}| = |\mathcal{V}|$.

Proof. From Lemma 4.1, it is known that $|\mathcal{K}_v|$ and $|\mathcal{V}_k|$ are constant. Since the system is unconditionally secure, then Bob can not learn anything about Alice's commitment before the right time. So, his tag $v \in \mathcal{V}$ can not reduce his uncertainty about x, which implies that $|\mathcal{K}_v| \geq |\mathcal{X}|$.

We can show that $|\mathcal{K}| \cdot |\mathcal{V}_k| = |\mathcal{K}_v| \cdot |\mathcal{V}|$ (see the proof for Corollary 3.1) and using Lemma 4.1, this brings $|\mathcal{K}| = |\mathcal{K}_v| \cdot |\mathcal{V}_k| \geq |\mathcal{X}| \cdot |\mathcal{V}_k|$. But since the system is optimal and the number of keys is minimal, then it must be that $|\mathcal{K}_v| = |\mathcal{X}|$. Since in an optimal system Alice is trusted as much as Bob, the success probabilities for each one's attacks must be equal. Then, $1/|\mathcal{X}| = 1/|\mathcal{V}_k| \Leftrightarrow |\mathcal{X}| = |\mathcal{V}_k|$ which implies $|\mathcal{K}| = |\mathcal{X}|^2 = |\mathcal{V}|$. □

Lemma 4.3 *If a commitment scheme $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ is optimal, then its incidence matrix can not be a BIBD.*

Proof. It was previously shown that $|\mathcal{V}| = |\mathcal{K}|$. Suppose the incidence matrix of the commitment scheme is a $2-(v,k,\lambda)$ Balanced Incomplete Block Design. Then the design is symmetric and by Theorem 2.14 in [1], any two keys have exactly λ tags validating them. But we have already seen that the maximum intersection between any two lines should be 1, so $\lambda = 1$. Then, by Theorem 2.1, each tag validates exactly $r = (v-1)/(k-1)$ keys. Again by Theorem 2.14 in [1], $r = k$. This means that $b = v = k^2 - k + 1$.

For any fixed value $y \in \mathcal{Y}$ each key generates one single value $x \in \mathcal{X}$. Since the system must be perfectly concealing, the keys validated by each tag should uniformly cover all possible y. By minimality, this means that $r = k = |\mathcal{Y}|$. By the same reasoning, the set of all keys must be a multiple of $|\mathcal{Y}|$. But this means that b must be a multiple of k. However, $b/k = k - 1 + 1/k$ and this cannot be divisible by $k > 1$. □

Lemma 4.4 Let $p = |\mathcal{X}|$. For a commitment scheme $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ that is optimal, the sum of distinct pairs of keys that don't have any tag in common is $p^2 \cdot (p-1)/2$.

Proof. To ease understanding, consider a square matrix where cell (i,j) indicates how many points there are in the intersection of key in row i and key in column j. The diagonal of the matrix is not filled.

Then, there are $p^2 \cdot (p^2 - 1)$ filled cells. Now we count the pairs that have a non-empty intersection. Each key has tag v_1 in common with $p-1$ different keys and since it is validated by p tags, each key contributes with $p(p-1)$ to the total of the sum

$$\sigma' = \sum_{k_i \in \mathcal{K}} \sum_{k_j \neq k_i \in \mathcal{K}} |\mathcal{V}_{k_i} \cap \mathcal{V}_{k_j}|$$

Therefore, $\sigma' = p^3(p-1)$. But this sum counts each pair twice, so the total number of distinct intersections is $\sigma = p^3(p-1)/2$.

We can find the total number of distinct key pairs that don't intersect, recalling that in this particular case all filled cells are either 0 or 1, which means that σ is the sum of all 1s in the table. As before, only a half of the matrix needs to be considered.

Then, the number of distinct key pairs without tags in common is

$$(p^4 - p^2)/2 - p^3(p-1)/2 = p^2 \cdot (p-1)/2$$

□

Theorem 4.1 *If a commitment scheme $CM(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{V}, f(k,x), g(v,k), \alpha, \beta)$ is optimal, then it is an affine resolvable commitment scheme, and all keys in each parallel class encrypt each value $x \in \mathcal{X}$ to the same value $y \in \mathcal{Y}$. That is, the function $y = f(k,x)$ depends only on the index of the parallel class containing k, and not on k itself.*

Proof. Let $p = |\mathcal{X}|$. From the previous results, it is known that there are p^2 keys. Fix some $x_0 \in \mathcal{X}$. The concealing property implies that there must be exactly p keys transforming x_0 into each possible value $y \in \mathcal{Y}$. Then, the keys can be grouped in p groups such that all keys $k_{i,j}$ in group i satisfy $f(k_{i,j}, x_0) = y_i$.

We know that there are exactly p keys validated by each verifier tag. For each two keys k_i and k_j validated by the same tag, it must happen that $f(k_i, x_0) \neq f(k_j, x_0)$ or else there won't be enough keys to hit all the values in $\mathcal{Y}$.

But by Lemma 4.4 and a counting argument, this implies that all pairs of keys in different groups must have exactly one common tag.

Since there are p disjoint keys in each group, each validated by p tags, each group forms a partition of $|\mathcal{V}|$ and is therefore a parallel class. The design is therefore resolvable and since the maximum intersection between two keys is $1 = k^2/v$ it is also affine.

Now consider a value $x_1 \in \mathcal{X}$ different from x_0. Suppose there are two keys k_i, k_j in different groups that code x_1 in the same way. That is:

$$\begin{aligned} &f(k_i, x_0) \neq f(k_j, x_0) \\ &f(k_i, x_1) = f(k_j, x_1) \\ &|\mathcal{V}_{k_i} \cap \mathcal{V}_{k_j}| = 1 \end{aligned}$$

Following the same reasoning as above, if $f(k_i, x_1) = f(k_j, x_1)$ then they can not have any tag in common, contradicting the previous division in groups. Therefore, keys in different groups code x in different ways and by a counting argument all keys in the same group transform x into the same y.

Repeating the argument for any $x_l \in \mathcal{X}$ and for all groups, it must happen that all keys k_i, k_j in the same group satisfy

$$f(k_i, x_l) = f(k_j, x_l)$$

and so the theorem is proved. □

Theorem 4.2 A resolvable design commitment scheme obtained from a resolvable $1-(v, k, \lambda)$ design $(\mathcal{D}, \mathcal{S})$ is a composition of a perfectly secure cipher system and an authentication code with $P_{d0} = k/v$ and $P_{d1} = \frac{\max |B_1 \cap B_2|}{k}$ for all $B_1, B_2 \in \mathcal{D}, B_1 \neq B_2$.

Proof. In the initialization of such a scheme, Alice is given a block B and Bob a point w such that $w \in B$. We assume that all parties involved know the encoding rules and the incidence matrix of the design in use. Then, B can be viewed as the index of the corresponding row in the matrix: a number between 0 and $r \cdot v/k - 1$. Likewise, w can be seen as a number between 0 and $v - 1$.

By definition of resolvable design, B belongs to some parallel class. Then, Alice's information can be written (i, j), where i is the index of the parallel class and j is the index of the block within the parallel class. The pair (i, j) can be interpreted as a pair source value / authenticator.

When Alice sends a commitment to Bob, she picks some $x_0 \in \mathbb{Z}_r$ and computes $y_0 = (x_0 + i) \mod r$. Then, x_0 can be seen as a symbol in alphabet $\Sigma = \mathbb{Z}_r$ and y_0 as a displacement of i positions in that alphabet. Effectively, Alice is applying a Ceasar's cipher to her secret message x_0. In general, Ceasar's cipher is not secure, but here, the message is composed of only one symbol, which means its size is equal to the size of the key. In this situation, it is equivalent to the One-Time Pad and is unconditionally secure. Note that all blocks in the same parallel class encipher x_0 in exactly the same way, which means that in fact, the parallel class alone represents the cipher key used by Alice.

Now, it remains to be seen that the design used to check the validity of the value revealed after the commitment can also be used to make an authentication scheme.

Draw the incidence matrix of the design used: each row represents a block and each column represents a point. Group the blocks of each parallel class together. Transposing the matrix, each block is now seen as a point in the authentication system, and because the original design was resolvable,

the new points can be divided in r groups of v/k elements with the property that each row has exactly one point in each group. Besides, each new point belongs to exactly k of the new blocks. In Appendix C, we illustrate with the example found in [3].

To view this design as an authentication code, it is enough to identify groups of points with a source value, and each point inside a group as an authenticator. The rows correspond to keys, and because each one has exactly one point in each group, each key computes exactly one authenticator for each source value.

The probability of an impersonation attack is the maximum probability of a given block containing Bob's w. Each point belongs to exactly k blocks. Since there are v blocks, the attack has probability k/v of succeeding.

For the substitution attack, suppose the attacker knows a block B_1 that contains Bob's point w. The attacker needs to find another block that contains w and belongs to a different parallel class.

Suppose the attacker picks a different block. He will succeed if any of the points in this block is w. For any other block B_2, let $d = |B_1 \cap B_2|$. Since every block has exactly k points, the chance of any of them being Bob's point is d/k.

Then, the probability of the substitution attack is $\frac{\max|B_1 \cap B_2|}{k}$ over all $B_1, B_2 \in \mathcal{D}, B_1 \neq B_2$. □

Appendix B: Flow Analysis

In the conclusion to the paper [3], the authors suggest a possible relation between commitment schemes and authentication schemes with arbitration, but point that the information flows between these systems are different.

Here, we analyse the different flows of information in a commitment scheme, and how these are realized through the flows present in the cipher and in the authentication systems. The following pictures help visualize the flows in the different systems.

In these figures, there are blocks representing each participant in the system and arrows representing the messages sent by them. Within some blocks is another name within square brackets. This represents the name of the user of the commitment scheme that will be playing the role indicated by the block. For instance, when Alice sends her commitment to Bob, he is playing the role of Eve in the cipher scheme: he receives a cipher text but can not read it.

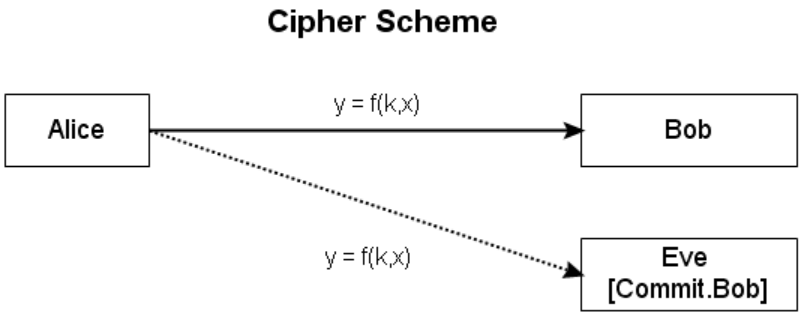

Figure 1: A Cipher Scheme

Next is shown how each flow is used to implement the flows of the final Commitment Scheme. We list the flows in each system according to the steps in the respective protocol. Some steps have two similar flows, and these are labeled 'a' and 'b'.

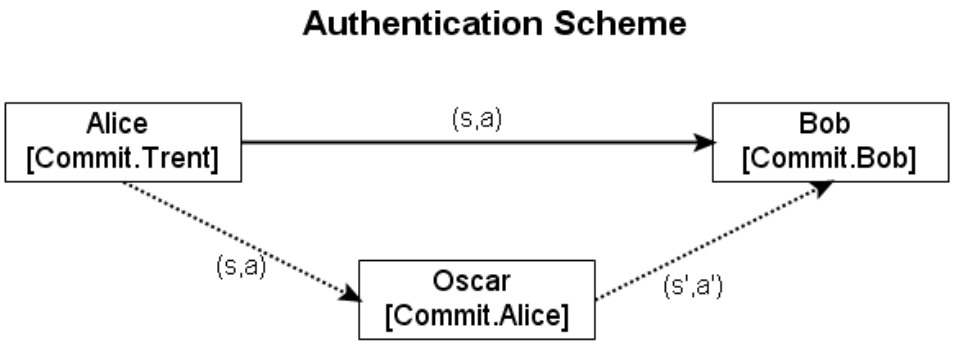

Figure 2: An Authentication Scheme

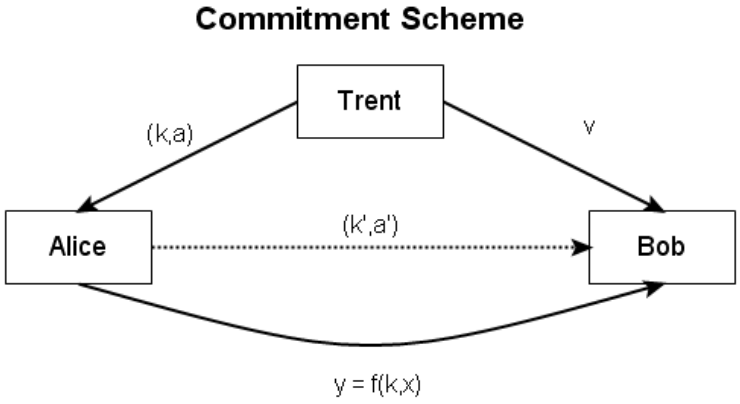

Figure 3: A Commitment Scheme

(CP1) Alice (a) and Bob (b) receive a secret key[1] by some secure channel.

(CP2) A message is sent from Alice to Bob (a) and possibly also read by Eve (b).

In an authentication system as described above, there are the following flows:

(AC1) Alice (a) and Bob (b) receive a secret key by some secure channel.

(AC2) A message is sent from Alice to Oscar (a), who may change it before relaying it to Bob (b)[2].

In a commitment scheme, there are the following information flows:

(CM1) Trent gives a key to Alice (a) and a verification tag to Bob (b).

(CM2) Alice sends her commitment to Bob.

(CM3) Alice sends her key to Bob to open her commitment.

The information flows of the commitment scheme are carried out by the information flows of the other systems like this:

- Flow (CM1.b) is achieved by flow (AC1.b). Flow (AC1.a) is ignored because Trent does not need to remember the key after he creates a valid message to send Alice. Flow (CM1.a) is achieved by flow (AC2.a), that is, Trent takes the role [AC.Alice] and sends a message to Alice ([AC.Oscar]). Due to the nature of the construction, flow (AC2.a) includes flow (CP1.a), because Alice now has a key for the cipher system.
- Flow (CM2) is achieved by flow (CP2.b).
- Flow (CM3) is achieved by flow (AC2.b). From this message, Bob deduces a key, completing flow (CP1.b), and opens the commitment by flow (CP2.a).

Cipher	**Commitment**	**Authentication**
(CP1.a)	(CM1.a)	(AC2.a)
...	(CM1.b)	(AC1.b) [(AC1.a)]
(CP2.b)	(CM2)	...
(CP1.b) (CP2.a)	(CM3)	(AC2.b)

Table 3: Information Flows

Appendix C: A Transposed Resolvable Design

The following tables show a resolvable $1-(6,2,5)$ design given in [3] and how it becomes a $1-(15,6,2)$ not resolvable design.

Table 4 shows how the points 0 to 6 are distributed by the 15 blocks. For this scheme, $\alpha = 1/3$ and $\beta = 1/2$.

Table 5 shows the transposed example.

[1] This includes the case where they create a key themselves and exchange it.

[2] This is just a simplified model. In reality, Alice sends the message to Bob, but Oscar may intercept and alter it or not.

Keys		Verifier Tags					
		0	**1**	**2**	**3**	**4**	**5**
	0	1					1
0	**1**		1			1	
	2			1	1		
	0		1				1
1	**1**	1		1			
	2				1	1	
	0			1			1
2	**1**		1		1		
	2	1				1	
	0				1		1
3	**1**			1		1	
	2	1	1				
	0					1	1
4	**1**	1			1		
	2		1	1			

Table 4: Original Example

	0			**1**			**2**			**3**			**4**		
	0	**1**	**2**	**0**	**1**	**2**	**0**	**1**	**2**	**0**	**1**	**2**	**0**	**1**	**2**
0	1				1				1			1		1	
1		1		1				1				1			1
2			1		1		1				1				1
3			1			1		1		1				1	
4		1				1			1		1		1		
5	1			1			1			1			1		

Table 5: Transposed Example

References

[1] Ian Anderson and Iiro Honkala, *A Short Course in Combinatorial Designs* http://www.utu.fi/ honkala/designs.ps Spring 1997.

[2] Birkhoff, G. and Mac Lane, S, *A Survey of Modern Algebra* 5th ed, p. 413, New York: Macmillan, 1996.

[3] C. Blundo, B. Masucci, D.R. Stinson, R. Wei, *Constructions and Bounds for Unconditionally Secure Non-Interactive Commitment Schemes* (Designs, Codes and Cryptography), Vol 26, 2002.

[4] Manuel Blum, *Coin flipping by telephone: a protocol for solving impossible problems* 24th IEEE Spring Computer Conference, pp 133 – 137 IEEE Press, 1982.

[5] Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[6] Ueli Maurer, *Authentication theory and hypothesis testing* IEEE Transactions on Information Theory, vol. 46(4), 1350-1356 2000.

[7] D. McGrew and J. Viega, *The Galois/Counter Mode of Operation (GCM)* Submission to NIST Modes of Operation Process, January, 2004.

[8] Ronald L. Rivest, *Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer* unpublished manuscript, http://citeseer.ifi.unizh.ch/rivest99unconditionally.html November 1999.

[9] Gustavus J. Simmons, *Authentication Theory / Coding Theory*, Advances in Cryptology: Proceedings of CRYPTO' 84, Lecture Notes in Computer Science, vol. 196, 411-432, Springer Verlag, Berlin, 1985.

[10] G. J. Simmons, *Message authentication: a game on hypergraphs* Congressus Numerantium, vol. 45, 161-192, 1984.

[11] Gustavus J. Simmons, *A Natural Taxonomy for Digital Information Authentication Schemes*, Advances in Cryptology - CRYPTO' 87: Proceedings, Lecture Notes in Computer Science, vol. 293, 269-288, Springer Verlag, Berlin, 1988.

[12] Douglas R. Stinson, *Combinatorial Characterization of Authentication Codes*, Advances in Cryptology: Proceedings of CRYPTO' 91, Lecture Notes in Computer Science, vol. 576, 62-72, Springer Verlag, Berlin, 1992.

[13] Douglas R. Stinson, *Some Constructions and Bounds for Authentication Codes*, Advances in Cryptology - CRYPTO' 86: Proceedings, Lecture Notes in Computer Science, vol. 263, 418-425, Springer Verlag, Berlin, 1987.