A Constructive Treatment of Authentication

Christopher Battarbee

School of Informatics, University of Edinburgh, United Kingdom

Abstract. We review the traditional techniques of message authentication from the perspective of Constructive Cryptography, one of the major composable security frameworks. For each of the 'textbook' means of message authentication, known results are compiled, and various gaps are addressed. We also point out areas of remaining study, and obstacles to demonstrating composability in those cases. The results can be thought of as a toolkit showing how to realise composable authentication with various cryptographic primitives, with precise statements on the security guarantees and set-up requirements.

1 Introduction

Classical cryptography can be thought of as providing two, somewhat orthogonal guarantees: privacy, and authenticity. In this article we address various means of achieving the latter, in a particular type of composable security framework.

Informally, authenticity is the process of binding data to a particular identity. As a motivating example, consider the one-time pad. We want to construct a system whereby honest parties A and B share a message, and an adversary E has no knowledge of this message. It is well-known that the one-time pad reveals no information about the underlying message to an adversary who is only given access to the ciphertexts – but without an authentication mechanism, B cannot confirm that data he receives originates from the intended communication partner, and so E can still violate the first of the two desired security properties by altering the ciphertext. In other words, the notion of privacy alone does not suffice to guarantee security against adversaries who can influence the communication, as well as observe it; that is, adversaries who behave *actively*.

In general, any cryptographic set-up in which an adversary can behave actively will require some means of authentication. One way to abstract this is to imagine that honest parties have access to an "authenticated channel", which behaves in a certain ideal way. Suppose we can show that a particular cryptographic mechanism realises this ideal authenticated channel; and that the whole system can be shown to be secure given access to an ideal authenticated channel. We would like to be able to conclude that the system using that particular authentication mechanism is secure, so that we can study each of these separately and avoid the complication that comes from studying the composite system.

This is the approach taken by Constructive Cryptography (CC), one of the two major composable security frameworks present in the literature (the other and probably better-known framework is Universal Composability, or UC). By

appealing in the standard way to a real-vs-ideal paradigm, the framework offers the composability outlined above in terms of the notion of *construction*: one starts with an ideal functionality and applies a protocol to transform it into a system that is indistinguishable from another ideal functionality, and we say that the latter ideal functionality is *constructed* from the former one. The composability then says that if resource B can be securely constructed from resource A, and resource C can be securely constructed from resource B, then resource C can be securely constructed from resource A. The modularity implicit from this result is of clear appeal for our purposes: we can study authentication in its own right, and as long as we can show that a particular authentication mechanism realises the appropriate notion of ideal authentication, we can infer the security of the whole system. Moreover, the ideal authentication can itself be constructed from various set-up resources (such as pre-shared keys), and so this approach allows us to study the set-up requirements and security profiles of systems when various different methods of realising authentication are applied.

All of this motivates the study of the textbook authentication protocols through the lens of the CC framework. In particular, the security of most of the authentication protocols in the literature is understood in terms of the advantage of attackers in security games; but satisfying such a definition does not automatically yield composable security. On the other hand, these protocols are already in widespread use, and their existing security bounds have already been thoroughly analysed, so we should aim for results that demonstrate that game-based security suffices for the composable security we target, if possible. Some results in this direction are known (if rather fragmented through the literature), but there remain several gaps in our understanding.

In this work, we continue the study of constructing authentication by means of the most well-known cryptographic mechanisms. We recall and codify results in this direction for universal hash functions (UHFs) and digital signature schemes (DSSs), and introduce new results pertaining to message authentication codes (MACs)and key-encapsulation mechanisms (KEMs) – this latter approach has received renewed attention in the context of post-quantum cryptography, in which the KEMs are for now considerably more efficient than their DSS counterparts. We introduce the standard game-based definitions of security, and show that the advantage in some (sometimes relatively strong) variant of that security bounds the advantage in distinguishing between the constructed and ideal resources, yielding composable security. Attention is paid to the type of set-up resources required, and the type of security yielded. We stress that where results along these lines were known they are compiled into a unified framework.

1.1 Related Work

We are not aware of a comparable document whose scope is to filter all means of achieving authentication through a composable framework. Perhaps closest to the spirit of our work is a report on how best to fulfil the authentication requirements for Quantum Key Distribution [Fre+24], which is also interested in defining authentication as some abstract functionality which can be fulfilled by

particular protocols, though this is only examined from a symmetric key perspective. Although the protocols are not afforded the same composable analysis we present in this work, the authors appear to share a fundamentally 'constructive' ethos; that authentication, is to be regarded as a module which can be plugged into a more advanced functionality, and can itself be realised with a variety of protocols.

This article quotes various results, with the aim of providing a complete picture of the composable landscape in authentication in a single document. These will be referred in the body of the work, but we highlight two papers of particular importance in which the basic ideal functionalities are defined [Por14a; MTC13].

Contrast the outlook of this article to a similar approach in UC: in that case we would define a protocol as a particular ideal functionality, and then show that this suffices to realise an ideal authentication functionality. Because protocols themselves are the ideal resources, rather than means of transforming ideal resources, it is arguably more difficult to isolate the precise set-up requirements in the UC framework. Some CC authors allude to this difference in perspective by describing CC as 'top-down', as opposed to UC, which is 'bottom-up'. Moreover, because CC is agnostic about its computational model (rather than defined explicitly in terms of various Turing machines), the authentication we construct here is easier to plug into quantum cryptography protocols, which often have an authentication requirement. Despite all this, the difference is to some extent a matter of taste, and a similar treatment would be possible in UC as well.

Finally, we direct the reader to various canonical references on the subject of Abstract, or Constructive Cryptography (henceforth CC). The framework is set out in a very high-level way by Maurer and Renner [MR11], but later receives a more grounded treatment [Mau10; Mau12]. The application of definitions and techniques in this work are closer in spirit to these latter two works. We supply a reasonably detailed overview of the necessary tools of CC for the purposes of self-containment; nevertheless, the reader is encouraged to treat [Mau10; Mau12] as reference works.

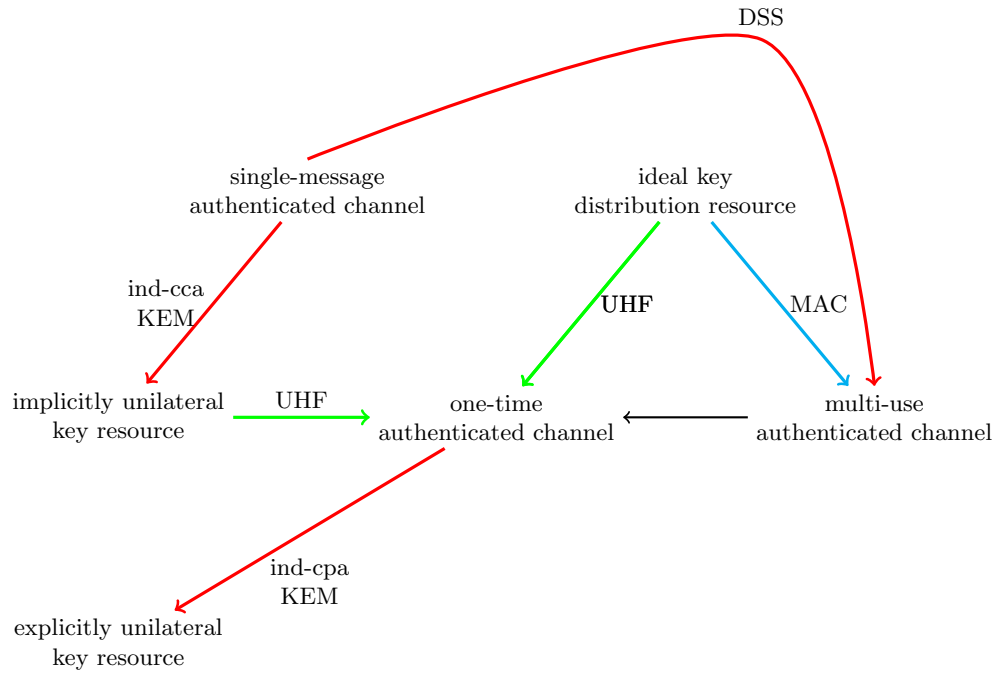
1.2 Results

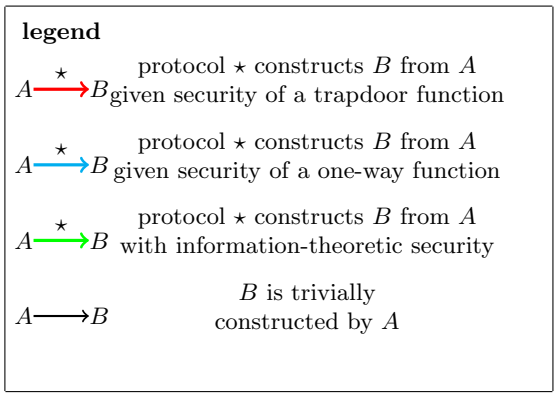
Our goal is to examine each of the 'textbook' authentication protocols through a constructive lens, thereby isolating their set-up requirements, the functionality they achieve, and the type of security they achieve. For the sake of readability a summary of each of the main results is presented here, with respect to this rubric.

- A universal hash function constructs ideal one-time authentication from a pre-shared key with unconditional security ([Por14a, Lemma 9])
- A message authentication code constructs ideal multi-use authentication from a pre-shared key with conditional security (Theorem 6)
- A digital signature constructs ideal multi-use authentication from one-time authentication with conditional security ([MTC13, Lemma 3])

– The combination of a KEM and a MAC construct ideal multi-use authentication from one-time authentication with conditional security (Corollary 4)

We summarise this state of affairs in Figure 1. Note that this is not an exhaustive list of the results in this paper.





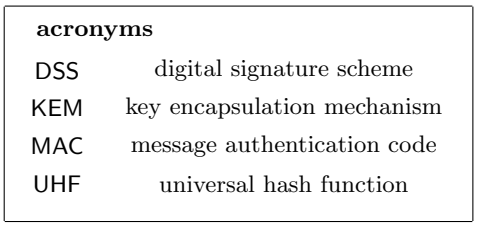


Fig. 1. A map of which functionalities can be constructed from which set-up assumptions, and with which security guarantees.

2 Preliminaries

2.1 Constructive Cryptography

We give an overview of the key ideas in Constructive Cryptography (CC), and introduce the idea of a reduction in this framework. This will allow us to prove that existing game-based notions of security suffice for composable security.

Systems The objects we consider in the abstract cryptography framework – be they ideal resources, or real systems constructed from resources and protocol converters – can all be thought of as taking input variables in an alphabet $\mathcal{X}$, and outputting variables in an alphabet $\mathcal{Y}$. Such a system is called an $(\mathcal{X}, \mathcal{Y})$ system. Let X_i be the random variable describing the ith input to the system, and Y_i be the random variable describing the ith output of the system. Given a particular symbol $y_i \in \mathcal{Y}$, the probability that $Y_i = y_i$ depends on the values of all input values $x_1, ..., x_n$, as well as previously outputted values $y_1, ..., y_n$. We also equip these distributions with *interfaces*, which are essentially a bookkeeping variable; from an interface set $\mathcal{I}$, each input and output x_i, y_i occurs at a particular interface $j \in \mathcal{I}$. A *system* is then the conditional probability distribution $\mathbb{P}(Y_i = y_i | X^i = x^i, Y^{i-1} = y^{i-1})$ together with the associated interface set. This description of a system's behaviour in minimal in the sense that if this distribution takes different values, it describes a distinct system.

Converters We can think of each interface of a system being available to a particular party (indeed we will abuse notation slightly and refer to a party's interface and a party by the same symbol). At this interface they receive data, to which they apply some cryptographic protocol, and submit the resulting data back at the same interface. We may then wish to argue about the behaviour of the new system after this transformation has been applied. This is formalised by the notion of a *converter*, which is a special type of system with a single *inner* interface, and a single *outer* interface. The inner interface is allowed to attach to the interface of the system, so that system outputs are now inputs to the converter; similarly, converter outputs are output to the system. Meanwhile, the outer interface now acts as the interface of a new system.

The $\{A, B, E\}$ Setting We are interested in the common setting of two honest parties with one malicious adversary, sometimes called the $\{A, B, E\}$[1] setting. Since every interface of the relevant systems corresponds to a party, every system has three interfaces. We only attach converters at the A and B interfaces, since A and B are the only parties applying cryptographic protocols.

[1] For Alice, Bob and Eve.

The Cryptographic Algebra In order to avoid excessive verbosity it will be useful to develop a notation that describes how systems are attached to each other. Attaching a converter π at interface $*$ of system S is denoted $\pi^* S$. This 'attaching' operation is independent of the order of composition, for example $\pi^A \rho^B S = \rho^B \pi^A S$ (where equality here means that the two conditional distributions defining the systems resulting from attaching the converters are the same).

Distinguishers, Simulators and Construction In the CC setting, we wish to show that it is impossible (or at least infeasible) to determine whether a system is an ideal resource, or the real system with a simulator attached, except with negligible probability. This notion is codified by the role of a distinguisher, which is given access to every interface of the candidate system. It may make adaptive queries to each of these interfaces (that is, it allowed to treat the conditional probability distribution defining the system as an oracle). Eventually, after submitting some finite number of queries, the distinguisher outputs a random variable Z that can take either 0 or 1. This variable is the distinguisher's guess of the system it interacts with. Since the distinguisher chooses its input query x_i with access to its previous choices of query, and the responses y_{i-1}, together with a distribution of its initial choice, the behaviour of the distinguisher is characterised by $\mathbb{P}(X_i = x_i | X^i = x^i, Y^{i-1} = Y^{i-1})$. In other words, the distinguisher is an $(\mathcal{Y}, \mathcal{X})$ system. Suppose that it halts after q queries; its behaviour is further characterised by $\mathbb{P}(Z = z | X^q = x^q, Y^q = y^q)$. Given two systems S, T, a distinguisher D's probability of correctly guessing which system it interacts with after q queries, minus $1/2$, is written $\Delta_q^D(S, T)$; we can also write this as $\Delta^D(S, T)$ if we are not concerned with the number of queries.

The ideal system and the real system may not have exactly the same syntax at the adversarial interface. For example, in the real world the adversary may be able to inject some message, and the cryptography ensures that invalid messages are rejected; whereas in the ideal world we simply do not allow the adversary to submit messages of its choice. We get around this with the notion of *simulation*; to the adversary's interface we attach a special resource called a *simulator*, which simulates the view of the adversary on the real system. The intuition is that if the ideal world can simulate the real world, in the sense that the distinguisher cannot tell the difference between the real world and the simulation, then the real world must be 'no less' secure than the ideal world.

Consider a special type of simulator called $\perp$, that just prevents any input or output at the adversarial interface. We introduce the notion of construction as follows. Let S, T be two ideal resources, π_1, π_2 be converters, and σ be a simulator. Suppose we have that

- $\pi_1^A \pi_2^B \perp^E S \equiv \perp^E S$
- For any distinguisher D,

$$\Delta^D(\pi_1^A \pi_2^B S, \sigma^E T) \leq \epsilon$$

for some $\epsilon > 0$

We say that π_1, π_2 ϵ-*construct* T from S, alternatively, we say the whole system $\pi_1^A \pi_2^B S$ ϵ-*realises* T . The first condition is called *availability*, and enforces correctness (usually it just follows from the correctness of the cryptographic mechanism). The second condition is called *security* and is generally where the work of proving comes in.

Finally, by way of notation, we use the arrow $\mapsto$ for construction: if converters π, ψ ϵ-construct resource T from resource S, we write

$$\pi^A \psi^B S \overset{\epsilon}{\longmapsto} T$$

Composition We are interested in two types of composition operation: the parallel composition of systems, and the sequential composition of converters.

The parallel composition of systems S and T, written $S||T$, is just the system obtained by taking the union of the interfaces available to each party. We obtain a new conditional probability distribution defining this new system. If we consider two systems to which a converter is attached, this operation allows us to specify the set-up resources for a cryptographic system (for example, the parallel composition of an insecure channel and a key distribution resource). Since converters are defined to have only a single inner interface, we need to be careful about how data from various system interfaces are handled at the inner converter interface, and it may be convenient to divide this inner interface into logical *sub*-interfaces.

The sequential composition of converters π, ψ, written $\pi \circ \psi$, is obtained by attaching the inner interface of converter π to the outer interface of converter ψ. This operation is not generally composable.

Importantly, we have a sequential composition theorem, which in our setting says that if π_1, π_2 ϵ_1-construct resource T from resource S, and ψ_1, ψ_2 ϵ_2-construct resource U from resource S, then the converters $\psi_1 \circ \pi_1$ and $\psi_2 \circ \pi_2$ $(\epsilon_1 + \epsilon_2)$-construct resource U from resource S. This is useful from both from an external and internal point of view. If we show a particular cryptographic mechanism constructs an authenticated channel, then any protocol requiring an authenticated channel as a set-up assumption is automatically secure if we use that cryptographic mechanism. On the other hand, we can analyse the construction of an authenticated channel in a modular way, for example by showing first that we construct secret keys, and then applying some cryptographic mechanism to these secret keys.

There is also a result on parallel composition, which is useful to guarantee security when many instances of the same protocol are run in parallel. Since we focus on the $\{A, B, E\}$ setting this will be less pertinent to our analysis.

Games as Systems Most existing cryptographic primitives are designed to fulfil game-based security requirements, which do not automatically yield composability. We would ideally like to show that these game-based notions suffice for composable security, which means we need a way to relate distinguishing

advantage to the advantage in one of these security games. The machinery for this is outlined below.

In traditional game-based security definitions, a challenger answers certain oracle requests for an adversary. This adversary terminates with some 'guess', which defines whether the adversary wins the game or not.

We can think of a game as a system, where oracle queries are inputs, and oracle answers are outputs. It is not generally convenient to define if the game is won based on data the adversary eventually produces, instead, we say that the game is won if the inputs outputs to the game satisfy certain conditions. This motivates an additional bookkeeping variable indicating on each system output if the game is won yet, called an 'MBO' (monotone binary output). Formally, our system outputs the usual variable Y_i as well as a variable A_i, which is set to 1 if the current transcript vectors X^i, Y^i satisfy the win condition of the game defined by the system, and 0 otherwise. This variable is monotone in the sense that once it is sent to 1, it cannot be then set to 0 again. Defining, in this fashion, the notion of 'winning' a game is a matter of convenience – note that an adversary to this game need not be given the MBO, because the probability of success is computed according to whether the transcript satisfies suitable conditions, or not. On the other hand, note that by adding an MBO we can turn any system into a system defining a game; so we call a system equipped with an MBO a 'game system'.

A distinguisher can still interact with a game system, but now, we are not interested in the output guess variable[2], but instead the following quantity. For a distinguisher D that halts after q queries, and a game system S, let A_i be the MBO defining the game from S. Define $\Gamma_q^W(S) = \mathbb{P}^{WS}(A_q = 1)$ to be the probability that the MBO of the system is set to 1 after q queries from the distinguisher W.

Two game systems are 'equivalent as games' if the probability distributions characterising their behaviour are the same on outputs for which the MBO variable is 0. In other words, two systems S, T are equivalent as games if, for $i \geq 1$, one has

$$\mathbb{P}^S(Y^i = y^i | X^i = x^i) = \mathbb{P}^T(Y^i = y^i | X^i = x^i)$$

for every pair of vectors y^i, x^i that do not define a winning event for the game implied by the system.

Critically, we have the following result of Maurer [Mau13, Lemma 2]. By way of notation we say that for a game system S, which necessarily comes with an MBO, the system S^- is the system defined by considering the game system without its MBO.

[2] The terminology 'distinguisher' is less helpful here, because we are not really interested in the probability that this entity can tell which distribution it is querying. For this reason, [Mau13] suggests referring to a distinguisher interacting with a game system as a 'game-winner'. We do not adopt this terminology as a matter of taste.

Theorem 1 ([Mau13, Lemma 2]). *Suppose two systems S, T are equivalent as games. For any distinguisher D, we have*

$$\Delta_q^D(S^-, T^-) \leq \Gamma_q^D(S)$$

Remark 1. This can be seen as a generalisation of the 'identical-until-bad' lemma of Bellare and Rogaway. To see this, note that we are effectively supposing that the systems behave identically unless some win condition occurs, and concluding that the 'difference' between these systems is no greater than the probability of that win condition.

In this framing, we have started with a system with an MBO, and considered the system created by removing that variable. In practice it might be more desirable to start with two generic systems whose distinguishing advantage we wish to bound, and equip them with an MBO such that we can apply Theorem 1. This motivates the idea of a reduction.

Reductions Reductions are a further tool of interfacing between distinguishability and games. In the case of a distinguishing game, this is rather natural: we start with the indistinguishability game and define a reduction that can interact with all the relevant oracles at its inner interfaces. At its outer interface it simulates a system to the distinguisher, so that from the distinguisher's perspective it interacts with the usual notion of a distinguisher; from the challenger's perspective, the composition of the distinguisher in the reduction behaves like the usual game-based notion of an adversary. Suppose we have that if the reduction composed with one case of the indistinguishability game behaves like (that is, describes the same distribution as) the real system, and the reduction composed with the other case behaves like the ideal system with a simulator attached at its adversarial interface. We can write the distinguishing advantage (by a standard argument) as

$$\frac{1}{2}|\mathbb{P}(D \text{ outputs } 1|\text{real system}) - \mathbb{P}(D \text{ outputs } 1|\text{ideal system})|$$

Since we can also think of the reduction as being composed with the distinguisher, if the distributions are the same the first term is the same as

$$\mathbb{P}(DR \text{ outputs } 1|\text{game chose } 0)$$

and similarly for the second term. It follows that the distinguishing advantage is the same as the advantage of the adversary defined by DR in the relevant indistinguishability game.

However, authentication related games generally do not have this decision format, instead requiring some 'passing' value to be submitted. The techniques are consequently more involved. In this regime, a reduction can be thought of as an intermediary layer between a generic system and a distinguisher that defines an MBO. In other words, if S is a system composed with a reduction

C, the resulting system $T = CS$ is such that $T^- = S$. Strictly speaking, the reduction has a single outer interface and a single inner interface, and mediates all interaction between the system and the distinguisher. This scenario is portrayed in Figure 2.

The general strategy to bound the distinguishing advantage of a real system S and an ideal resource T is then roughly as follows:

1. Construct a simulator for the ideal resource
2. Define a reduction that equips each candidate system with an MBO
3. Show that the resulting game systems are identical-as-games and apply Theorem 1 to show that for any distinguisher $\mathcal{D}$, $\Delta_q^{\mathcal{D}}(T, S) \leq \Gamma_q^{\mathcal{D}}(CS)$
4. Show that a distinguisher interacting with the system with an MBO produces a transcript that defines a winning game (equivalently, causes the MBO to be set to 1 upon halting) with probability equal to the advantage of an adversary playing the relevant security game.

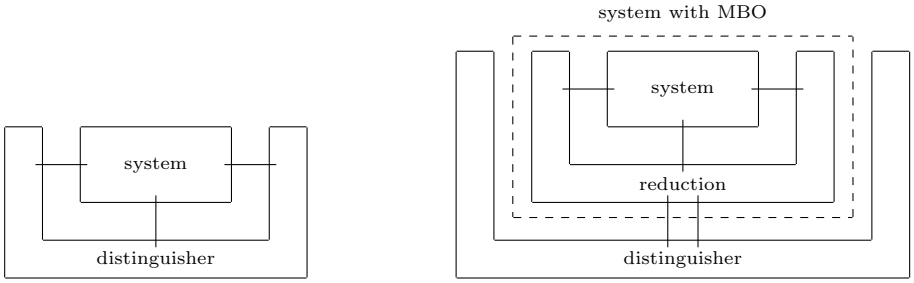


Fig. 2. *Left:* a generic system interacting with a distinguisher. *Right:* a generic system with an intermediary reduction layer. The part of the system within the dashed lines can be understood to behave as a system with an MBO.

Conditional vs. unconditional security It is sometimes possible to guarantee that the probability of a distinguisher's success is independent of its computational power, in which case we have *unconditional* or *information-theoretic* security. If security is instead based on the presumed intractibility of some computational problem, it is customary to define the advantage in a security game asymptotically; that is, as a function of some security parameter. In this case, since we relate distinguishing advantage to the advantage in a security game, the distinguishing advantage will also be asymptotic, i.e., a function. If the security parameter is denoted κ, the distinguishing advantage is written $\Delta_q^D(\kappa)$.

Remark 2. A party who is in possession of a key that is believed to be known only to the two honest parties can trivially violate authenticity. The requirement that the ideal key-sharing resource only sends its contents to the intended parties

is therefore necessary. Moreover, in the case of multiple partners, one requires a separate ideal key-sharing resource with each (and, implicitly, one has to keep track of which ideally-shared keys correspond to which identity).

2.2 Ideal Resources

We cover the two ideal authentication resources we wish to construct throughout this work, with a one-time and multi-use variant.

Suppose that an honest party wishes to transmit classical information to another honest party, in the presence of a third, adversarial party who controls the channel between them. Such an adversary can always drop the communication by 'cutting' the channel, but should otherwise learn precisely the classical data being forwarded down this channel, and no more. In particular the receiving party should output either exactly the data that was input by their intended partner, or an error. In CC terms, we define two resources with this intuition in mind: first, a one-time authenticated channel, depicted in Figure 3.

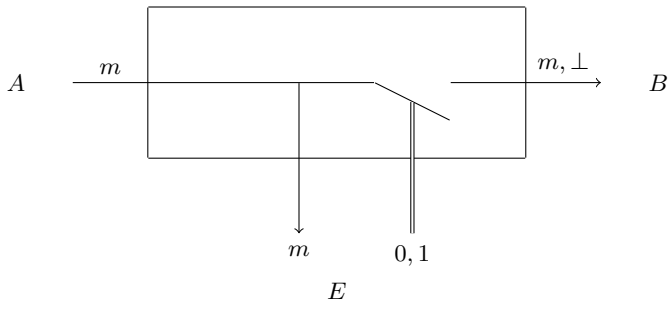


Fig. 3. An ideal one-time authenticated channel.

First codified in [Por14b], this resource captures the notion that the only thing B can receive is something that was input by the intended communication partner, or an error. Moreover, we allow the adversary E to see the contents of the message. On the other hand, this functionality only captures one-time authentication – there is no provision for the input of multiple messages by A.

In order to capture a multi-use channel, [MTC13] introduces the notion of a buffer as a property of the channel. Every time A inputs some classical data, it is stored in the (unbounded) buffer. We now allow the adversary E to input messages of her choice to the channel, but these are only output to B if they were previously input by A to the buffer. We thereby capture the notion of the unavoidable ability of an adversary in control of the channel to choose the order in which messages are delivered – but we do not allow them to insert messages of their choice. As before, we also let the adversary read the contents of any data input by A. In this case B only receives messages that were input by A,

but not necessarily in the right order, or an error. This functionality is depicted in Figure 4.

Remark 3. Unlike with our ideal one-time authenticated channel (and indeed, unlike in [MTC13]), we do not provide the adversary with a switch to cut the channel. The reason is that we are modelling an adversary who can choose whether or not to allow messages input by A to pass through the channel by injecting them in any order; but we do not require that they inject all, or even, any of the messages input by A. In other words, the controls provided to the adversary suffice to emulate the switch cutting the channel.

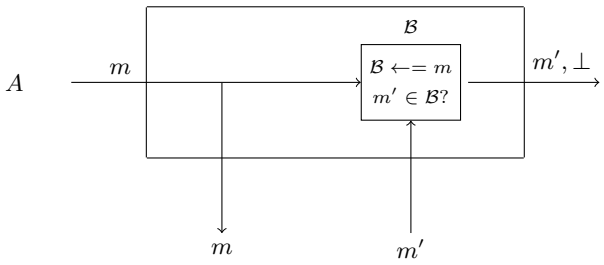


Fig. 4. An ideal multi-use authenticated channel.

Set-up Resources. Recall the notion of construction, whereby we attach protocols to some ideal resource, such that the resulting system is indistinguishable from another ideal resource. In this way we can codify what our set-up assumptions are: they are the ideal resources we use to construct our authenticated channels. The two resources of note are the ideal key distribution resource (Figure 17), corresponding to pre-shared keys; and a one-time authenticated channel (Figure 15), corresponding to receiving an authentic copy of the appropriate public key. Given that we also seek to construct the one-time authenticated channel this might seem a bit circular - we defer an analysis of this objection to later on in the chapter.

Bounded vs. Unbounded Use Channels As well as the one-time authenticated channel we could talk about a q-time authenticated channel, for some $q > 1$. Notice that we can always construct such a channel from q parallel instances of a system constructing a one-time authenticated channel. The multi-use authenticated channel we have discussed, by contrast, is implicitly supposed to admit an unlimited number of uses. As we will see in Corollary 2, we can not hope for unconditionally secure unbounded use-channels in the message-tag paradigm; in other words, for a specified integer q it is possible to construct an unconditionally secure q-time authenticated channel, but if we wish to assume an unbounded number of uses on the channel we can only hope for security against computationally bounded distinguishers.

Identities. Implicitly, the ideal authenticated channels are parametrised by the identities of Alice and Bob. In particular, the ideal functionality guarantees that anything received by Bob can only have been sent by Alice – if Bob had multiple communication partners, he would require multiple ideal channel resources. Compare this to the ideal authentication functionality defined in the UC framework [Can04], where the ideal channel is a single resource to which parties explicitly submit their own identity, as well as the identity of the intended partner, along with every message. Since both formulations are hypothetical the choice between them is a matter of taste; we have opted for the former.

Glossary of Set-up Resources. We list here all the notation used for ideal resources that are assumed or constructed in our security proofs, as a reference for parsing statements in the cryptographic algebra. Formal definitions are resources are found in Appendix A.

- $(\mathcal{C}^*)^{\mathcal{A}\to\mathcal{B}}$ - the insecure channel from A to B. Anything A submits into this resource is output at E's interface, and anything output at B's interfaces was input at E's interface. We can similarly define $(\mathcal{C}^*)^{\mathcal{A}\leftarrow\mathcal{B}}$, $(\mathcal{C}^*)^{\mathcal{A}\leftrightarrow\mathcal{B}}$
- $\mathsf{kd}^{A\leftrightarrow B}$ - the ideal key distribution resource, implicitly parametrised by a key-space $\mathcal{K}$ and a key generation algorithm (which by default is uniform sampling). The resource selects a key from its keyspace and outputs it at the A and B interfaces, and outputs nothing at the E interface.
- $\mathsf{ot}-\mathsf{ac}^{A\to B}$ - the one-time authenticated channel from A to B. A inputs a message m at her interface, which is output at the adversarial interface E. E can choose whether or not to deny service on the channel; if she does, B gets an error $\perp$, if she does not, B gets A's message m.
- $\mathsf{m}-\mathsf{ac}^{A\to B}$ - the multi-use authenticated channel from A to B. A can input an unbounded number of messages to the resource, which are output to the adversary and stored in an internal buffer. E can input messages of her choice to the resource, but they are output to B if and only if they are already in the buffer. In particular, E can only cause messages input by A to be output at B, but can cause any such messages to appear in any order
- $\mathsf{ot}-\mathsf{sc}^{A\to B}$ - the one-time secure channel from $A \to B$. A inputs a message m, and its length $|m|$ is output at E. E can choose whether or not to deny service on the channel; if she does, B gets an error $\perp$, if she does not, B gets A's message m.
- $\mathsf{m}-\mathsf{sc}^{A\to B}$ - the one-time secure channel from $A \to B$. A can input an unbounded number of messages, the length of each of which is output to E. If j messages have been input by A, for any $i < j$ E can submit i at her interface and cause the ith message sent by A to be output at B. This is the same functionality as the multi-use authenticated channel, whereby E can cause messages from A to B to appear in any order.

2.3 Protocols

The Message-Tag Paradigm. All the protocols we discuss achieve authenticity by appending some auxiliary tag to a message, where the message is sent in

the clear. Upon receipt of a message-tag pair, an honest party wishing to verify the authenticity of the message evaluates a particular function on the message and tag, which has a binary output: either 'accept' or 'reject'. We call this the *message-tag paradigm*. One can show that authenticity is achieved, in the sense of constructing the ideal resources described above, if it is not possible for a dishonest party to produce message-tag pairs that are evaluated as 'accept' by this function. This is the heart of the idea of 'unforgeability', which underpins all authenticity security guarantees, but has various flavours across the different settings. We are not aware of any types of cryptographic authentication that do not follow this approach of appending a tag to a message.

Symmetric keys vs. asymmetric keys. We can split the protocols into those that require a key that must remain secret to the two communication parties; and those that generate two keys, one of which can be shared arbitrarily (but must be delivered to communciation partners) authentically.

Having set the scene, we are now ready to examine methods of classical authentication. In the rest of this chapter we will review the security of various protocols for classical message authentication, with attention to three factors: the required set-up resources, the functionality achieved, and the type of security realised. We call these the symmetric key setting, and the public key setting (the latter is sometimes known as the asymmetric key setting). The report will be divided into the separate study of each regime.

3 Symmetric Key Setting

Our first task is to discuss methods of classical message authentication when we assume the honest parties possess a shared, private key.

3.1 Standard Authentication

In the most general sense within the symmetric key setting, and following the message-tag paradigm we have set out in the introduction, we can imagine that two parties, Alice and Bob wish to append tags from a space $\mathcal{T}$ to messages from a space $\mathcal{M}$. Suppose that two parties share a private function $F : \mathcal{M} \to \mathcal{T}$, selected uniformly at random from some family of functions $\mathcal{F}$. When Alice wants to convince Bob of the authenticity of classical data m, she sends m together with $t = F(m)$. When Bob receives some pair $(m', t') \in \mathcal{M} \times \mathcal{T}$, he can verify whether or not $F(m') = t'$; if it does, he is convinced of the authenticity of the classical data m'.

Before formalising this approach, we first note that any of the secret functions we discuss above must be selected uniformly at random from a family of such functions. It is convenient for such a family to be indexed by some auxiliary set, such that we can treat members of this auxiliary set as private keys. The following establishes this setting.

Definition 1. *Let $A = \{0, ..., |A| - 1\}$ and $\{0, ..., |B| - 1\}$ be sets of integers such that $|A| \geq |B|$. A family of hash functions is a set $\mathcal{H} = \{h : h \in f(A, B)\}$, parametrised by some set K.*

Notice that, since we must have $|K| = |\mathcal{H}|$, the size of K is bounded by the total number of functions from A to B. On the other hand, $\mathcal{H}$ need not consist of all such functions.

The protocol associated to such a family of functions is known variously in the literature – we go with the 'standard authentication' terminology of [Por14b]. It is defined as follows:

Definition 2 (Standard Authentication). *Let $\mathcal{H}$ be a family of hash functions mapping from a message space $\mathcal{M}$ to a tag space $\mathcal{T}$. Let K be the associated parametrising set of $\mathcal{H}$. The standard authentication protocol is as follows:*

- KeyGen *samples an element of K uniformly at random and outputs it.*
- Mac_1 *takes as input a private key $k \in K$, a message $m \in \mathcal{M}$, and outputs a tag $t = h_k(m)$.*
- Vfy *takes as input a private key $k \in K$, a message-tag pair $(m, t) \in \mathcal{M} \times \mathcal{T}$, and checks if $h_k(m) = t$. If it does, it outputs m, otherwise it outputs $\perp$.*

The reason for using the Mac_1 terminology will become clearer later on.

Strong Universality and the Security of Standard Authentication In [WC81] (following the introduction of the type of object considered therein in [CW77]), and in the adjunct paper [Sti94], sufficient conditions are given for a family of hash functions to yield (non-composable) security in the standard authentication protocol. These conditions are known as the ϵ-almost strong universality[3] of a family of functions; for brevity we will sometimes shorten this to simply 'strong universality'.

Definition 3. *Let $\mathcal{H}$ be a family of hash functions from A to B. For $0 < \varepsilon < 1$, the family of hash functions is ϵ-almost strongly universal if*

1. *For every $x \in A, y \in B$, one has*

$$|\{h \in \mathcal{H} : h(x) = y\}| = \frac{|\mathcal{H}|}{|B|}$$

2. *For every $x_1, x_2 \in A, y_1, y_2 \in B$,*

$$|\{h \in \mathcal{H} : h(x_1) = y_1, h(x_2) = y_2\}| \leq \frac{\epsilon|\mathcal{H}|}{|B|}$$

[3] When they were introduced, these type of hash functions were presented as possessing a property called 'universality', as opposed to strong universality. It was later realised that additional conditions were needed for the security of authentication schemes derived from these functions, hence 'strongly'; on the other hand, it turns out that the original examples of functions also satisfy strong universality.

In [WC81], only $\epsilon = 1/|B|$ is defined. Indeed, this value is optimal, and if it is obtained one checks the second condition implies the first condition. The relaxation of the second condition to include non-optimal bounds, in which case the first condition is not immediate, is Stinson's treatment [Sti94].

The most famous property of strongly universal hash functions is that they can yield information-theoretic authenticity. The traditional arguments for the sufficiency of strong universality in guaranteeing information-theoretic unforgeability of an associated standard authentication protocol, which are given essentially in [WC81] and [Sti94], are roughly as follows. Suppose some adversarial party who has no information about the secret key k, but has seen exactly one valid message-tag pair under h_k, is trying to come up with a different message-tag pair that is valid under h_k. Intuitively, the first condition of strong universality encodes that, from a single valid message-tag pair, one learns only negligible information about which function was used to generate the tag. The second condition encodes that, given a valid message-tag pair, there are a very small number of message-tag pairs that are valid under the same function. One can then demonstrate that it is difficult to come up with a valid message-tag pair without knowledge of the secret function directly.

In order to justify this formally, we should show that any adversary has negligible advantage in an appropriately defined security game. However, this would not immediately yield *composable* security. In fact, the composable security of standard authentication is proved by a direct probabilistic argument without the need to define a security game (a situation that will change as we go forward). As such we eschew such technicalities, and move directly to composable security.

Composable Security Recall that the composable treatment, instead of describing how the protocol interacts with some security game, is to describe how the protocol is constructed as a system, and then to show that this resulting system is distributed almost identically to an ideal system. We are in the symmetric key setting, so we assume some ideal key-distribution resource; and since authenticity is only a relevant concern in the presence of an active adversary, we assume classical data is transmitted via an insecure channel.

The standard authentication system is then constructed as a parallel composition of an ideal key-sharing resource, and an insecure channel resource, with converters running the standard authentication protocol attached at the honest parties' interfaces.

Definition 4. *Let $\mathcal{H} : \mathcal{M} \to \mathcal{T}$ be a family of functions, with K the associated parameterising set, and let $(\mathsf{KeyGen}, \mathsf{Mac}_1, \mathsf{Vfy})$ be the algorithms of the associated standard authentication protocol. The standard authentication system with respect to $\mathcal{H}$, depicted in Figure 5, is the parallel composition of an ideal key-sharing resource, and an insecure channel resource, where the ideal key resource runs KeyGen. Attached to Alice's interface is a converter with a running Mac_1, and attached to Bob's interface is a converter running Vfy. We call these converters $\pi_{\mathsf{tag}}^{\mathsf{s-auth}}, \pi_{\mathsf{vfy}}^{\mathsf{s-auth}}$, respectively.*

In the language of the cryptographic algebra, this system is denoted

$$\left(\pi_{\mathsf{tag}}^{\mathsf{s-auth}}\right)^{A}\left(\pi_{\mathsf{vfy}}^{\mathsf{s-auth}}\right)^{B}\left((\mathcal{C}^{*})^{A\to B}||\mathsf{kd}^{A\leftrightarrow B}\right)$$

Remark 4. The converter attached to Alice's interface has one input at its outside interface, and one input and one output at its inside interface. Bob's converter, meanwhile, has two inputs at its inside interface, and one output at its outside interface. Notice that the system is therefore well-defined, since the converters have the same number of inputs and outputs as the algorithms of the standard authentication protocol.

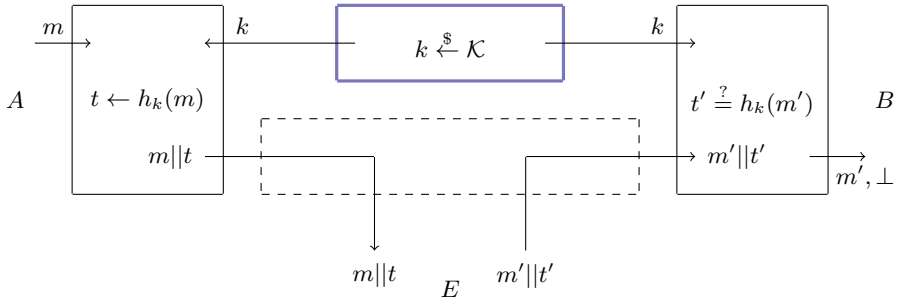


Fig. 5. The standard authentication system, constructed from an ideal key-sharing resource, and an insecure channel.

It turns out that strong universality of the family of hash functions used in the standard authentication system also suffices for the composable security of that system. Interestingly, this was only recently demonstrated by Portmann [Por14b], some 30 years after the introduction of strong universality.

Theorem 2 ([Por14b, Lemma 9]). *A standard authentication system constructed from an ϵ-almost strongly universal family of hash functions ϵ-realises an ideal one-time authenticated channel in the presence of an unbounded distinguisher. In the language of the cryptographic algebra,*

$$\left(\pi_{\mathsf{tag}}^{\mathsf{s-auth}}\right)^{A}\left(\pi_{\mathsf{vfy}}^{\mathsf{s-auth}}\right)^{B}\left((\mathcal{C}^{*})^{A\to B}||\mathsf{kd}^{A\leftrightarrow B}\right)\overset{\epsilon}{\longmapsto}\mathsf{ot}-\mathsf{ac}^{A\to B}$$

Necessity of Strong Universality Portmann's argument demonstrates that strong universality is sufficient for realising a one-time authenticated channel. On the way to doing so, he notes that, if $P_{XTX'T'Y}$ is the joint distribution for the real system of the message input and honest tag output, the message-tag pair input at the adversarial interface, and the output at Bob's interface (thus, a distribution completely describing the behaviour of the real system), the probability of distinguishing the two systems is bounded by

$$\sum_{t\in\mathcal{T}}\sum_{x,x't'\in\mathcal{M}^2\times\mathcal{T}} P_X(x)P_{X'T'|XT}(x',t'|x,t)\mathbb{P}_{k\xleftarrow{\$}K}\left(h_k(x)=t\wedge h_k(x')=t'\right)$$

The rightmost term in the summand is $\epsilon/|\mathcal{T}|$ by the definition of strong universality. For each t, the quantity $P_X(x)P_{X'T'|XT}(x',t'|x,t)$ describes the probability of the distinguisher choosing a particular message input x, and forgery attempt $x'||t'$, for a tuple $(x,x',t)\in\mathcal{M}^2\times\mathcal{T}$ – so, summing over all these tuples give 1. The double sum is therefore equal to $\sum_{t\in\mathcal{T}}\epsilon/|\mathcal{T}|$, whence the result follows.

We point out the following consequence of this argument, which to our knowledge has not been set down before. Suppose that we have constructed a standard authentication system from some family of functions $F:\mathcal{M}\to\mathcal{T}$, and we are given that this family of functions ϵ-realises a one-time ideal authenticated channel. The statistical distance between the two distributions must then be bounded by ϵ, and exactly Portmann's argument shows that we must have

$$\sum_{t\in\mathcal{T}}\sum_{x,x't'\in\mathcal{M}^2\times\mathcal{T}} P_X(x)P_{X'T'|XT}(x',t'|x,t)\mathbb{P}_{f\xleftarrow{\$}F}\left(f(x)=t\wedge f(x')=t'\right)\leq\epsilon$$

In other words, *any* family of functions that ϵ-realises a one-time authenticated channel is tightly restricted on the possible value of

$$\mathbb{P}_{f\xleftarrow{\$}F}\left(f(x)=t\wedge f(x')=t'\right)$$

for each (x,t,x',t'). With the restriction that this value should be constant on all such quadruples, one recovers the second condition of ϵ-almost strong universality (note that it turns out the first condition is neither sufficient, nor necessary, for the composable security of standard authentication). This can be seen as a partial converse to the assertion that a standard authentication system constructed from a strongly universal family of functions realises an ideal authenticated channel.

Bounds on Key Length Notice that, since we insist a family of hash functions is parameterised by a set K, the number of required keys is the number of required functions. We can therefore discuss key length in terms of the logarithm of the minimum required number of hash functions to achieve strong universality.

Intuitively, the minimum size of $|\mathcal{H}|$ should be a function of both $|\mathcal{M}|$ and $|\mathcal{T}|$: $|\mathcal{H}|$ cannot be too small compared to $|\mathcal{M}|$, or too many messages would be mapped to the same tag; nor can it be too small compared to $\mathcal{T}$, or not every tag would be in the range of the functions in $\mathcal{H}$. In fact, the size of $|\mathcal{T}|$ dominates, as the following corollary to a result of Stinson [Sti94, Theorem 6.2] demonstrates.

Theorem 3 ([Sti94, Theorem 6.2]). *Let $\mathcal{M}$ be a space of messages such that $\mathcal{M}=2^a$, and $\mathcal{T}=2^b$. There exists a $1/2^{b-1}$-almost strongly universal family of hash functions from $\mathcal{M}$ to $\mathcal{T}$, such that $\log|\mathcal{H}|=(i+1)s+b$, where*

– $s = b + \lceil \log \log a \rceil$
– $i = \log(a'/s)$

The quantity $(i+1)s+b$ is roughly $(b+\log\log a)\log a$. In other words, given a message space $\mathcal{M}$ and a tag space $\mathcal{T}$, the length of keys required to $1/|\mathcal{T}|$-realise an ideal authenticated channel is the length of the tags, plus some very slow-growing factor in the length of the messages. In some sense, this result demonstrates that unconditional authenticity requires keys that are of a length that, in a practical sense, is independent of the length of message to be authenticated. Compare this remarkable state of affairs favourably to the unconditionally secure scheme that ensures privacy, as opposed to authenticity – the one-time pad. In that case, Shannon's theorem tells us that the length of the keys is lower-bounded by the length of the messages. In other words, as far as unconditional security is concerned, there is an extremely stark discrepancy between resource requirements in the privacy and authenticity contexts. We will return to this subject when we discuss Quantum Key Expansion.

Following our discussion in Section 2.2, we can combine this result with the parallel composition theorem to obtain the following:

Corollary 1. *For $q \in \mathbb{N}$, one can realise an ideal q-time authenticated channel on a message space of size $|\mathcal{M}|$ with keys of length $\mathcal{O}(q \log(|\mathcal{M}|))$.*

Key Reuse We have shown that universal hash functions construct a one-time authenticated channel. On the other hand, in some sense, if the keys were drawn from a larger space (and were therefore longer), one might expect to be able to realise multiple-use authenticated channels. In fact, this relationship between key-size and reusability is understood quite generally, due to [WC81]. In the following, note that since the key space indexes the set of available hash functions, the size of these two sets is the same.

Theorem 4 ([WC81, Theorem, p.10]). *Suppose a family of hash functions $\mathcal{H}$ is such that no unbounded adversary can compute a valid message-tag pair with probability greater than some $0 < p < 1$, after seeing n message-tag pairs. We must have $|\mathcal{H}| > 1/p^n$.*

This result is not automatically in the composable setting, but nevertheless yields a useful corollary for our purposes.

Corollary 2. *There is no family of hash functions such that the associated standard authentication system realises an unbounded-use ideal authenticated channel with bounded probability, with respect to an unbounded distinguisher.*

Proof. Consider the contrapositive to Theorem 4: if there are fewer than $1/p^n$ hash functions to be sampled from, there exists an adversary that can compute a valid message-tag pair with advantage exceeding p. This adversary's advantage in distinguishing a real resource and an ideal resource would then also exceed p, since it could submit a message-tag pair it knows would be rejected by an ideal

multi-use channel, and guess that it interacts with an ideal system if and only if that message-tag pair is accepted by the system. This strategy succeeds with probability at least that of successfully computing a forged message-tag pair. In other words, this adversary can be converted into a distinguisher that violates the second property of realising the ideal resource, and the result follows.

It is worth noting that [WC81] suggests a method of reusing the same hash function; one introduces more secret key material and XORs it with the computed tag on each application of the hash function. It turns out that this does leak some information about the utilised hash function, but in a rather controlled and slow way, as demonstrated by [Por14b].

Examples of Strongly Universal Hash Functions In general, strongly universal hash functions can be naturally constructed by appealing to modular arithmetic, roughly because taking residues distributes inputs 'evenly' across an output space in a manner sufficient for strong universality.

The classical construction, introduced by Wegman and Carter in the same paper that introduces the notion of universality [CW77], goes as follows. In order to construct a family of strongly universal hash functions between a set $A = \{0, 1, ..., |A|-1\}$ and $\{0, ..., |B|-1\}$, with $|A| > |B|$, choose a prime $p > |A|$. With $|B|$, the family of hash functions

$$\{h_{x,y}(m) = ((mx + y) \bmod p) \bmod |B|\}$$

turns out to achieve strong universality. This is an example of a family of hash functions that, when first introduced, were proven only to be universal. When strong universality was introduced in [WC81], it turned out that the construction satisfied the stronger conditions as well. Many other families of functions, and means of constructing such functions, are also given across two papers.

As an instructive example, let us see a family of optimally strongly universal hash functions introduced by Stinson.

Theorem 5 ([Sti94, Theorem 5.2]). *Let q be a prime number. For integers $s > t$, consider any surjective linear transformation $\phi : \mathbb{F}_q^s \to \mathbb{F}_q^t$, recalling that $\mathbb{F}_q^s$ and $\mathbb{F}_q^t$ can be seen as algebras over $\mathbb{F}_q$ of dimension s and t, respectively. The family of hash functions from $\mathbb{F}_q^s$ to $\mathbb{F}_q^t$, defined by*

$$\mathcal{H} = \{h_{x,y}(m) = \phi(xm) + y : x \in \mathbb{F}_q^s, y \in \mathbb{F}_q^t\}$$

is $1/q^t$-almost strongly universal.

A derivation of this construction gives rise to the example referenced by Theorem 3.

3.2 Message Authentication Codes

In practical terms, we might find that the restrictions on key re-use for universal hashing are too cumbersome. After all, the assumption of an ideal key-sharing

resource in the standard authentication system is a strong one, requiring physical contact, or trusted couriers, to realise – ideally, we would like to be able to appeal to this resource only once. However, we have seen from our Corollary 2 that we cannot hope to authenticate an unbounded number of messages with unconditional security. In this section, we study those authentication methods with computational security, and in particular, the notion of a message authentication code.

Definition 5. *A message authentication code* MAC *is a trio of algorithms* $(\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Vfy})$. *Associated to these algorithms are a key space* $\mathcal{K}$*, a message space* $\mathcal{M}$*, and a tag space* $\mathcal{T}$*, consisting of all the admissible keys, messages and tags, respectively. The algorithms are defined as follows:*

- KeyGen *takes a security parameter* κ *as an input, and outputs a private key* $k \in \mathcal{K}$
- MAC *is a deterministic algorithm that takes as input a private key* $k \in \mathcal{K}$ *and a message input* $m \in \mathcal{M}$*, and outputs a tag* $t \in \mathcal{T}$*.*
- Vfy *takes as input a key* $k \in \mathcal{K}$*, a message-tag pair* $(m, t) \in \mathcal{M} \times \mathcal{T}$*, and outputs either* $\perp$ *(reject) or the message* m*.*

Specifying the spaces $\mathcal{K}, \mathcal{M}, \mathcal{T}$ specifies whether the MAC is allowed to take variable-length messages, or output variable-length tags, and so on. Note that since we have insisted Vfy is deterministic, in practice the algorithm Vfy computes a tag on the received message using MAC and the secret key, and compares this to the received tag. Finally, we have defined the verification algorithm to output the message, rather than the usual 'accept' notification usually found in the literature. The definitions are clearly equivalent, but ours is slightly more convenient from a composable standpoint.

Security of Message Authentication Codes Broadly speaking, we say a MAC is secure if it is difficult to *forge* tags, i.e. one should not be able to produce passing message-tag pairs without knowledge of the secret key. We would also like this to hold even if an adversary can see valid tags of messages of its choice, which we model by allowing the adversary to query a tagging oracle. Eventually, the adversary will submit some message-tag pair, and be told whether or not this pair is valid under the challenger's verification function. One can think of this as being given a single query to a verification oracle; an obvious generalisation is to consider the case where multiple queries to a verification oracle are permitted, and saying that the MAC is insecure if any of these queries are answered affirmatively.

Since MACs are deterministic, one could trivially win this type of unforgeability game by obtaining a valid tag on some message from the tagging oracle, and submitting the resulting message-tag pair to the verification oracle. This scenario is directly forbidden in the security definition – in fact, submitting a message-tag pair that has already been generated is known as a *replay attack*, and it is standard to treat these separately. As such, security is defined in terms

of the inability to forge valid tags on 'fresh' messages. We might, however, be interested in a stronger property: the additional inability to forge fresh tags on messages for which a valid tag has already been seen. This stronger variant could also allow single, or multiple, queries to a verification oracle, so we have four variants of unforgeability.

With this in mind, security is characterised by the following functions. We use the modular bookkeeping [BGM04]; a paper whose results on the relationship between these notions we will return to presently.

Definition 6. *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Vfy})$ *be a* MAC *with message space* $\mathcal{M}$ *and tag space* $\mathcal{T}$*, and let* $\mathcal{A}$ *be an adversary. Let* $\mathsf{Exp}^{\mathsf{type}}_{\Pi,\mathcal{A}}(\kappa)$ *be one of the games described in Figure 6, where* $\mathsf{type} \in \{\mathsf{UF}-1, \mathsf{UF}, \mathsf{SF}-1, \mathsf{SF}\}$*. The advantage of* $\mathcal{A}$ *in the* type*-unforgeability game against* Π*, denoted* $\mathsf{Adv}^{\mathsf{type}}_{\mathcal{A},\Pi}(\kappa)$*, is a function of the security parameter, defined by*

$$\mathsf{Adv}^{\mathsf{type}}_{\mathcal{A},\Pi}(n) = \mathbb{P}\left(\mathsf{Exp}^{\mathsf{type}}_{\Pi,\mathcal{A}}(\kappa) = 1\right)$$

Algorithm 1 Game $\mathsf{Exp}^{\mathrm{type}}_{\Pi,\mathcal{A}}(\kappa)$

$k \leftarrow \mathsf{KeyGen}(\kappa)$
$\mathsf{TagSet} \leftarrow \{m : \{\}\}$ for $m \in \mathcal{M}$
$\mathsf{f} \leftarrow 0$; $\mathsf{sf} \leftarrow 0$; $\mathsf{count} \leftarrow 0$
$\mathcal{A}^{\mathsf{Tag}(),\mathsf{Ver}()}$
if $\mathsf{type} = \mathsf{UF}-1$ **then**
    return $f \wedge (\mathsf{count} \stackrel{?}{=} 1)$
else if $\mathsf{type} = \mathsf{UF}$ **then**
    return f
else if $\mathsf{type} = \mathsf{SUF}-1$ **then**
    return $\mathsf{sf} \wedge (\mathsf{count} \stackrel{?}{=} 1)$
else if $\mathsf{type} = \mathsf{SUF}$ **then**
    return sf
end if

Algorithm 2 Oracle $\mathsf{Tag}(m)$

$t \leftarrow \mathsf{Mac}(k, m)$
$\mathsf{TagSet}[m] \leftarrow \mathsf{TagSet}[m] \cup \{t\}$
return t

Algorithm 3 Oracle $\mathsf{Ver}(m, t)$

$d \leftarrow \mathsf{Vfy}(k, m, t)$
if $d \stackrel{?}{=} m$ **then**
    if $\mathsf{TagSet}[m] \stackrel{?}{=} \emptyset$ **then**
        $\mathsf{f} \leftarrow 1$
    end if
    if $t \notin \mathsf{TagSet}[m]$ **then**
        $\mathsf{sf} \leftarrow 1$
    end if
end if
$\mathsf{count} \leftarrow \mathsf{count} + 1$
return d

Fig. 6. The various types of MAC unforgeability game.

In plain English, the game maintains a dictionary in which the keys are all possible messages, and the values are sets. Implicitly, this is initialised such that an empty set corresponds to every possible message, though in practice

the dictionary can be generated on-the-fly. When a message is submitted to the Tag oracle, the resulting tag is added to the list corresponding to the submitted message. When a message-tag pair is submitted to the verification oracle, if the list corresponding to the submitted message is empty, the flag f is set to 1 to signify that the usual unforgeability notion is violated; if the corresponding list is non-empty, we set the sf flag to 1 if it does not contain the submitted tag, to signify that the strong unforgeability notion is violated. Depending on the particular version of the experiment, the game may also verify that the counter is 1, to enforce that a single verification query is made.

Remark 5. In the 'standard' notion of MAC security, when the adversary is run it is assumed to produce a message-tag pair as its attempted forgery, ending the experiment. Unlike in this standard scenario, in our definition of $\mathsf{UF}-1$ security, an adversary can continue to query its tagging oracle after it has generated its forgery. However, since only a single verification query is allowed it gains no advantage from doing so, so indeed $\mathsf{UF}-1$-security is equivalent to the standard notion of MAC security.

Remark 6. The type of unforgeability we define, whereby an adversary can choose its own message to sign, is sometimes called *existential* unforgeability.

Relationship between MAC *unforgeability notions.* The relationship between the four types of security defined in Definition 6 is given in [BGM04]. Crucially, the authors demonstrate (contrary to prevailing popular belief) that $\mathsf{UF}-1$ security does not imply UF security (though the converse holds). $\mathsf{SUF}-1$-security, meanwhile, does suffice for UF security, though the two notions are not equivalent.

Composable Security To argue composably about the security of a MAC, we employ a similar set-up to the previous section. Along these lines, let us first define a MAC system.

Definition 7. *Let* $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Vfy})$ *be a message authentication code with key space* $\mathcal{K}$*, message space* $\mathcal{M}$*, and tag space* $\mathcal{T}$*. A* MAC *system (which depends on the choice of* MAC*), depicted in Figure 7, is the parallel composition of an ideal key-sharing resource, and an insecure channel resource, where the ideal key resource runs* KeyGen*. Attached to Alice's interface is a converter* $\pi_{\mathsf{tag}}^{\mathsf{MAC}}$ *with a single input port at its outside interface, running* Mac*, and attached to Bob's interface is a converter* $\pi_{\mathsf{tag}}^{\mathsf{MAC}}$ *with a single output port at its outside interface, running* Vfy*.*

Remark 7. This time, notice that the key-sharing resource depends on a security parameter. Implicitly we consider that a different resource is defined for each choice of parameter. Again, notice that the inputs and outputs of the converters are such that the system is well-defined.

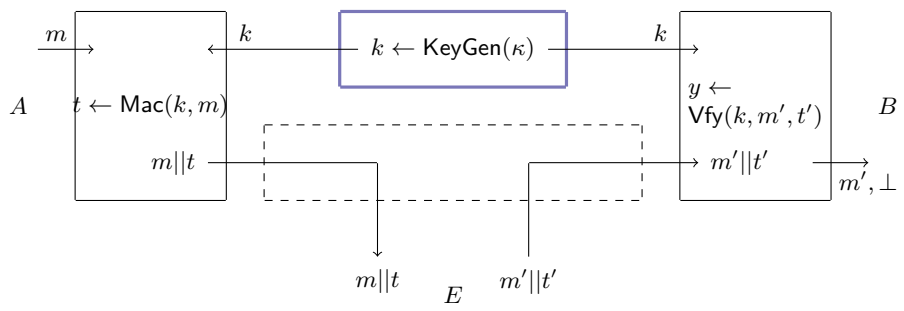


Fig. 7. A MAC system, constructed from an ideal key-sharing resource, and an insecure channel.

We are going to argue that a MAC system ϵ-realises a multi-use ideal authentication channel, where ϵ is a bounded adversary's advantage in the MAC security game. To do so we argue similarly to [MTC13, Lemma 2]; that is, by plugging the candidate system into a reduction that allows us to bound the distinguishing advantage of a bounded distinguisher by the success probability in the appropriate security game. We call upon the technical ingredient [Mau13, Lemma 2] to facilitate this argument, which is a result that can be seen as the composable version of the famous 'identical-until-bad' lemma. This is an application of known ideas in the area, but to our knowledge has not been recorded before.

Theorem 6. *Let Π be a* MAC*, and consider the associated* MAC *system. For any bounded distinguisher, there exists an bounded adversary $\mathcal{A}$ such that the* MAC *system $\epsilon(\kappa)$-realises an ideal multi-use authenticated channel in the presence of a computationally bounded distinguisher, where $\epsilon(\kappa) = \mathsf{Adv}^{\mathsf{uf}}_{\mathcal{A},\Pi}(\kappa)$. In the language of the cryptographic algebra,*

$$(\pi^{\mathsf{MAC}}_{\mathsf{tag}})^A(\pi^{\mathsf{MAC}}_{\mathsf{vfy}})^B\left((\mathcal{C}^*)^{A\to B}||\mathsf{kd}^{A\leftrightarrow B}\right) \overset{\epsilon(\kappa)}{\longmapsto} \mathsf{m}-\mathsf{ac}^{A\to B}$$

Proof. The availability condition follows immediately from the correctness of the MAC. It remains to show that the real system, and an ideal system with a simulator attached at the adversarial interface (depicted in Figure 8), are infeasible to distinguish for a bounded distinguisher.

The simulator, which is assumed to have been constructed with the appropriate value of the security parameter, behaves as follows. It generates its own choice of key from the key generation algorithm, and whenever it receives a message m at its inside interface, it computes $t = \mathsf{Mac}(k, m)$ on this message, and outputs $m||t$ at its outside interface. Whenever the distinguisher input a message-tag pair $m'||t'$, the simulation checks whether $t' = \mathsf{Mac}(m')$. If it does, m' is input at the adversarial input port; otherwise, do nothing (recall from our discussion of the multi-use ideal channel that one can think of this as implicitly activating a killswitch).

If a filter is attached to Eve's interface, the two systems clearly behave identically. It remains to bound the advantage of a distinguisher when it is allowed to interact with the adversarial interface.

In this direction, we construct a reduction that works as follows. When the reduction receives a tagging oracle query it inputs that query at the Alice interface of the candidate system, and collects the resulting tag at the output port of Eve's interface. When the reduction receives a verification oracle query, it inputs the message-pair at the input port of Eve's interface, and collects the output at Bob's interface. The reduction also needs to keep track of the messages that have been submitted, and the tags that have been queried, is detailed in the unforgeability game.

Clearly, the real system and the ideal system generate tags according to the same distribution. We therefore just need to check that every time a verification oracle query response defines a losing game, the output of the response has the same distribution across both systems.

There are two ways that a verification oracle query can lose: either the message component was previously submitted, in which case the reduction doing the bookkeeping will always record the game as losing; or the message component of the queried message-tag pair was never input as a tagging query, but the message-tag pair fails the verification check. Let us examine these cases separately.

Let (m, t) the message-tag pair submitted to the verification oracle, and suppose that a tagging query has previously been submitted on m. The probability that the output is $\perp$ is $\mathbb{P}_{k \leftarrow \mathsf{KeyGen}(\kappa)}(\mathsf{Mac}(k, m) \neq t)$ for both systems. Since, if the verification is passed, the ideal system will always output m at Bob's interface, since by assumption it was previously input at Alice's interface, the probability the output is m is also the same for both systems.

If a tagging query has not previously been submitted on m, we only need to consider the scenario that the verification is not passed (otherwise we would define a winning game) - but as we noted in the previous case, this is the same across both systems.

The real and ideal systems, when equipped with an MBO by the reduction, are therefore equivalent as games. Any distinguisher's advantage in guessing which system it interacts with is therefore bounded by the probability that the transcript that that distinguisher outputs defines a win condition. To conclude it suffices to show that such a distinguisher is equivalent to an adversary playing the uf unforgeability game – but this is immediate, since an adversary playing such a game has exactly the same oracles to query, and receives outputs with exactly the same distribution.

Since MACs are assumed only to have security against computationally bounded adversaries, we realise the multi-use authenticated channel only in the presence of computationally bounded distinguishers. Notice also that the 'usual' notion of MAC security ($\mathsf{uf} - 1$ security, in our formulation) does *not* suffice for the proof to go through – instead we require the stronger notion of uf security, where the number of verification queries can exceed 1. The reason is that a distinguisher

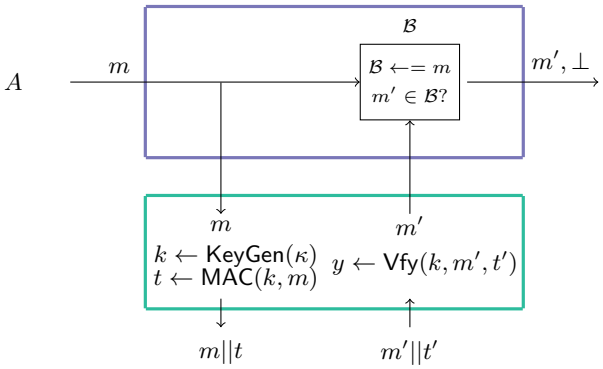


Fig. 8. The ideal multi-use authenticated resource, with a simulator attached at the adversarial interface.

gets access to all interfaces of the system, in particular the input port of the adversarial interface, and so automatically can have more than 1 verification query. This seems to be an unavoidable obstacle, since any multi-use channel should allow multiple inputs from the adversary – on the other hand, per the discussion of [BGM04], allowing multiple verification queries is in some sense the 'correct' formulation of MAC formulation anyway.

Examples There are a great many MAC schemes considered computationally secure, based on various primitives, including: universal hash functions[4] [Bla+99], cryptographic hash functions [ST08], and block ciphers [IK03; BKR00]. From our point of view, we are primarily concerned with whether these algorithms are unconditionally secure or not, and so a complete survey of proposed computationally secure message authentication codes is well out of scope for us. The references we provide should be understood to be only a very small sample of the complete literature on the subject.

Because the security of MACs are based on computational assumptions, and these assumptions are supposed to include the ability of the MACs to tag an unbounded number of messages, there are not strict theoretical restrictions on the sizes of keys, or the number of safe uses of these keys. Compare this to the situation with strongly universal hash functions, where unconditionally ϵ-realising the ideal resource imposed lower bounds on key length, and upper bounds on the number of times this key could be used. Once computational assumptions are introduced, the key length is more of a consequence of design choices, where shorter keys may be possible for different families of MACs. In general, it is not surprising to note that the performance of computationally secure MACs is significantly better than that of strongly universal hash functions, since the

[4] These are message authentication schemes built with universal hash functions that do not provide unconditional security, as in standard authentication. Some authors do not treat them as pure MACs, since they are not deterministic.

motivation for switching to computational assumptions is the impracticality of managing a large amount of key material.

On the subject of security, we note that the stronger notion of security required for our composable security result to hold does not appear to be an obstacle in practice, since all the schemes we have mentioned in this section are known to be uf secure, as demonstrated by [BGM04].

Finally, we remark that the presence of pre-shared keys allows one to temper the 'reasonableness' of the computational assumptions made, in the sense that it is possible to construct MACs whose security depends on the feasibility of inverting certain candidate one-way functions. We will have more to say about this when we evaluate the different candidates for message authentication.

3.3 Authenticated Encryption

When application data is being exchanged in the symmetric-key context, in the most general case we are interested in both privacy and authenticity. It is a folklore result that we can achieve this dual security, called *authenticated encryption*, by first encrypting with a CPA-secure encryption scheme, and computing a tag on this ciphertext with a MAC. In fact, the constructive paradigm is well-equipped to recover a composable variant of this result, as we now demonstrate.

We first have to define precisely what we mean a CPA-secure encryption scheme.

Definition 8. *An encryption scheme* E *is a trio of algorithms* $\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$. *Associated to these algorithms are a symmetric key space* $\mathcal{K}$*, a message space* $\mathcal{M}$*, and a ciphertext space* $\mathcal{C}$*, consisting of all the admissible symmetric keys, messages, and ciphertexts respectively. The algorithms are defined as follows:*

- KeyGen *takes a security parameter* κ *as in input, and outputs a key* k *from the space* $\mathcal{K}$
- Enc *takes as input a symmetric key* k *and a message* m*, and returns a ciphertext* c
- Dec *takes as input a symmetric key* k *and a ciphertext* c*, and returns a message* m

The encryption scheme is said to be correct if for any symmetric key k we have $\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$. Its security is defined by the game depicted in Figure 9; intuitively, the adversary picks a message, and the challenger decides at random whether to encrypt that message, or a random string. The adversary wins if it can guess which of its choices was encrypted; it also gets the power to see encryptions on messages of its choice. Note that this necessitates the encryption being non-deterministic.

Definition 9. *Let* $\mathsf{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme, and let* $\mathcal{A}$ *be an adversary. Let* $\mathsf{Exp}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$ *be the game described in Figure 9. The advantage of* $\mathcal{A}$ *in the* $\mathsf{ind-cpa}$*-indistinguishability game against* E*, denoted* $\mathsf{Adv}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$*, is a function of the security parameter, defined by*

$$\mathsf{Adv}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa) = \mathbb{P}\left(\mathsf{Exp}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)\right) - \frac{1}{2}$$

This notion of security was introduced as [Bel+97, Definition 2], together with a related notion known as 'left-or-right' security (as opposed to this notion of 'real-or-random' security). Left-or-right security is the notion given by some standard textbooks, notably [KL07] – we opt for real-or-random because it fits more neatly into a composable framework. The point is somewhat moot since, from [Bel+97], the two notions can be shown to be equivalent.

Algorithm 4 Game $\mathsf{Exp}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$

$k \leftarrow \mathsf{KeyGen}(\kappa)$
$m^* \leftarrow \mathsf{Adv}()$
$b \xleftarrow{\$} \{0,1\}$
$r \xleftarrow{\$} \mathcal{M}$
if $b = 0$ **then**
  $c^* \leftarrow \mathsf{Enc}(k, m^*)$
else
  $c^* \leftarrow \mathsf{Enc}(k, r)$
end if
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}()}(c^*)$
return $b' \stackrel{?}{=} b$

Algorithm 5 Oracle $\mathsf{Enc}(m)$

return $\mathsf{Enc}(k, m)$

Fig. 9. The indistinguishability game for encryption schemes, in the presence of chosen-plaintext attacks.

Because we want to construct a multi-use secure channel we will need a variant of this notion of security with multiple challenges.

Definition 10. *Let* $\mathsf{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme, and let* $\mathcal{A}$ *be an adversary. Consider a variant of the game described in Figure 9 where the adversary submits* n *messages, and either these* n *messages or encrypted, or* n *random strings are encrypted. The challenger answers* cpa *queries as normal. We call this game* $\mathsf{Exp}^{n-\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$*, and denote its advantage*

$$\mathsf{Adv}^{n-\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa) = \mathbb{P}\left(\mathsf{Exp}^{n-\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)\right) - \frac{1}{2}$$

Composable Security We proceed according to an argument similar to that put forward in [CMT13], which we will return to later. As set out in Section 2.1, we can use reductions to translate between the language of games and distinguishers. As usual we start by defining the construction.

Definition 11. *Let* $\mathsf{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme with key space* $\mathcal{K}$*, message space* $\mathcal{M}$*, and ciphertext space* $\mathcal{C}$*. An AE system is the parallel composition of a multi-use authenticated channel and a key distribution resource*

with key space $\mathcal{K}$. Attached to Alice's interface is a converter $\pi^{\mathsf{E}}_{\mathsf{Enc}}$ with a single input at its outside interface, at which it receives messages m. At its inside interface it receives a key k from the key distribution resource, and runs Enc *with respect to (k, m); it also outputs the resulting ciphertext c to the authenticated channel. Attached to Bob's interface is a converter $\pi^{\mathsf{E}}_{\mathsf{Dec}}$ with a single output at its outside interface. At its inside interface it receives a key k from the key distribution resource, and a ciphertext c (which may take the value $\perp$). It runs* Dec *with respect to (k, c), and outputs the resulting m (which may take the value $\perp$) at its outside interface.*

We have the following.

Theorem 7. *Let* E *be an encryption scheme, and considered the associated AE system. Suppose n messages are to be submitted to this channel. For any bounded distinguisher, there exists a bounded adversary $\mathcal{A}$ such that the AE system $\epsilon(\kappa)$-realises an ideal multi-use secure channel in the presence of this distinguisher, where $\epsilon(\kappa) = n\mathsf{Adv}^{\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$. In the language of the cryptographic algebra,*

$$(\pi^{\mathsf{E}}_{\mathsf{Enc}})^A(\pi^{\mathsf{E}}_{\mathsf{Dec}})^B \left(\mathsf{m} - \mathsf{ac}^{A\to B} || \mathsf{kd}^{A\leftrightarrow B}\right) \stackrel{\epsilon(\kappa)}{\longmapsto} \mathsf{m} - \mathsf{sec}^{A\to B}$$

Proof. We aim to bound the advantage of a distinguisher by the advantage of an adversary in the $\mathsf{Exp}^{n-\mathsf{ind-cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$ game. To do so we proceed according to the following strategy: we define a simulator on the ideal system, and a reduction, such that the reduction composed with the game behaves like the real world in one case of the game, and the ideal world with a simulator in the other case of the game. As we have discussed in Section 2.1, this suffices to recover the bound.

We first define the simulator. The simulator runs KeyGen to obtain an internal value k. At its internal interface the simulator receives the lengths of the messages input at the A interface from the multi-use secure channel; when it does, it samples a string of that length and encrypts it with k, outputting the resulting ciphertext c_i. It also stores each c_i it computes; when it receives c', if c' matches one of the internally held values c_i, the simulator submits i at its internal interface.

The reduction has to handle n message inputs from the distinguisher, and has to produce n corresponding ciphertexts. It also has to handle n ciphertext inputs, and produce n corresponding message outputs. To do this it submits the n message inputs to the game, which in turn produces n ciphertexts; these are output at the E interface of the reduction. It handles all the decryption internally, without appealing to the game; when the distinguisher submits c' at E's interface, if $c' = c_i$ for one of the ciphertexts generated by the game, output the corresponding m_i at B's interface, otherwise output $\perp$.

If the game chose to encrypt the real values, the output at E's interface is distributed as in the AE system. Moreover, because the buffer of the multi-use authenticated channel in the AE system will only accept ciphertexts generated by A's converter, the behaviour is also the same at B's interface. If the game chose to encrypt random values, the output at E's interface is distributed as the

ideal system composed with the simulator. Since the simulator keeps track of which ciphertexts it has generated, and rejects any ciphertext input not already stored by the simulator, the behaviour at B's interface.

This argument that shows that the distinguishing advantage is bounded by $\mathsf{Adv}^{n-\mathsf{ind}-\mathsf{cpa}}_{\mathsf{E},\mathcal{A}}(\kappa)$. To conclude with the claim of the theorem we apply the union bound of [BBM00].

Remark 8. Strictly speaking we have not constructed an unbounded-use secure channel here, because we have to know *a priori* a bound on the number of messages that the channel will be required to process in order to compute the security bound. In practice we might say that we have a channel that can tolerate a 'polynomial' number of uses.

4 Public-Key Setting

Supposing that one does not wish to go to the trouble of pre-sharing private, symmetric keys, an alternative solution is provided by public-key, or asymmetric, cryptography[5].

The idea of using public-key cryptography for authentication was put forward at the same time as the discovery of the field itself, in the seminal work of Diffie and Hellman [DH22]. The early work in this area is concerned with digital signature schemes, though as we shall see, there are alternative primitives in public-key cryptography which are appropriate for authentication.

4.1 Signatures

Continuing our examination of authentication schemes within the message-tag paradigm, we arrive at the notion of a digital signature, which is by far the most popular method of ensuring classical message authentication without symmetric keys. Let $\mathcal{M}$ be a space of messages, and $\mathcal{T}$ be a space of tags. In contrast to the symmetric case, we imagine one party generates two functions, say (F, G), such that F is used to assign elements of $\mathcal{T}$ to elements of $\mathcal{M}$, and G is used to determine whether a pair[6] $(m, \sigma) \in \mathcal{M} \times \mathcal{T}$ represents a valid assignment. Crucially, the function G can be made public, hence 'public-key' cryptography (the difference in privacy requirements between F and G have also earned this approach the title of 'asymmetric' cryptography).

More formally, we have the following.

[5] In the context of quantum communication, which is sometimes advertised as quantum-*safe* communication, it is common to describe only *post-quantum cryptography* in this context. Post-quantum cryptography is a subset of public-key cryptography, and the recently standardised primitives have precisely the same functionality as those we describe in this section. As such, we use the public-key, rather than post quantum, terminology, for the sake of generality, and defer an analysis of the necessity of enforcing 'post-quantum' to the Discussion section.

[6] It is common in the literature to use σ, for 'signature', in the public-key context.

Definition 12. *A digital signature scheme Σ is a trio of algorithms* KeyGen, $\mathsf{Sign}, \mathsf{Vfy}$. *Associated to these algorithms are a key space $\mathcal{K}_s \times \mathcal{K}_v$, a message space $\mathcal{M}$, and a signature space $\mathcal{T}$, consisting of all the admissible keys, messages and signatures, respectively. The algorithms are defined as follows:*

- KeyGen *takes a security parameter κ, and outputs a key pair (sk, vk) (sometimes called the 'signing key' and the 'verification key', respectively)*
- Sign *is a (not necessarily deterministic) algorithm that takes as input a signing key $sk \in \mathcal{K}_s$ and a message input $m \in \mathcal{M}$, and outputs a signature $\sigma \in \mathcal{T}$*
- Vfy *takes as input a verification key $vk \in \mathcal{K}_v$, a message-signature pair $(m, \sigma) \in \mathcal{M} \times \mathcal{T}$, and outputs either $\perp$ (reject) or the message m.*

Again, we have defined the verification algorithm to output the message, rather than 'accept'.

Security of Digital Signatures The security game for digital signatures is similarly defined to the ones defining the security of MACs, and was put forward as the 'right' notion of signature security in [GMR88]. Although the usual expository treatment, and the one we have adopted here, is to discuss first symmetric key, then public key authentication, the notion of unforgeability was first defined for digital signatures. However, because the verification key is public, we may suppose that any adversary has access to unlimited verification queries. As such, it no longer makes sense to define the security game in terms of the number of requests made to a verification oracle, so we return to the traditional method of defining such games, and define the success probability of the adversary relative to some particular message-signature pair it produces. On the other hand, we retain the requirement that forgeries must be on fresh messages.

Definition 13. *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ be a digital signature, and let $\mathsf{Exp}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa)$ be the experiment described in Figure 10. The advantage of $\mathcal{A}$ in the $\mathsf{euf-cma}$-unforgeability game against Σ, denoted $\mathsf{Adv}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa)$, is a function of the security parameter, defined by*

$$\mathsf{Adv}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa) = \mathbb{P}\left(\mathsf{Exp}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa) = 1\right)$$

Composable Security Let us see that a digital signature can construct an ideal multi-use authenticated channel from a single-use authenticated channel, and an insecure channel, in the sense that these two resources cannot be distinguished with probability exceeding the advantage of that distinguisher in the $\mathsf{euf-cma}$-unforgeability game. To begin, we define the construction.

Definition 14. *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ be a digital signature with key space $\mathcal{K}_S \times \mathcal{K}_v$, message space $\mathcal{M}$, and signature space $\mathcal{T}$. A signature system, depicted in Figure 11 is the parallel composition of a one-time authenticated channel, and an insecure channel resource, parametrised by κ. Attached to Alice's interface is a*

Algorithm 6 Game $\mathsf{Exp}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa)$

$(sk, vk) \leftarrow \mathsf{KeyGen}(\kappa)$
$\mathsf{Q} \leftarrow \{\}$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}()}$
if $\mathsf{Vfy}(vk, m^*, \sigma^*) \stackrel{?}{=} m^* \wedge m \notin \mathsf{Q}$ **then**
  return 1
else
  return 0
end if

Algorithm 7 Oracle $\mathsf{Tag}(m)$

$\sigma \leftarrow \mathsf{Sign}(sk, m)$
$\mathsf{Q} \leftarrow \mathsf{Q} \cup \{m\}$
return σ

Fig. 10. The unforgeability game for digital signatures.

converter $\pi^{\Sigma}_{\mathsf{sgn}}$, with a single input port at its outside interface, that runs KeyGen *once and inputs vk to the ideal authenticated channel. Whenever the converter receives an input at its outside interface, it computes a signature on this message with the singing key it generated, and forwards the message-signature pair down the insecure channel. Attached to Bob's interface is a converter π^{Σ}_{vfy} which receives a value vk, which it records. Whenever it receives a message-signature pair from the insecure channel, it runs* Vfy *on this pair with respect to the verification key it received, and outputs the result at its outside interface.*

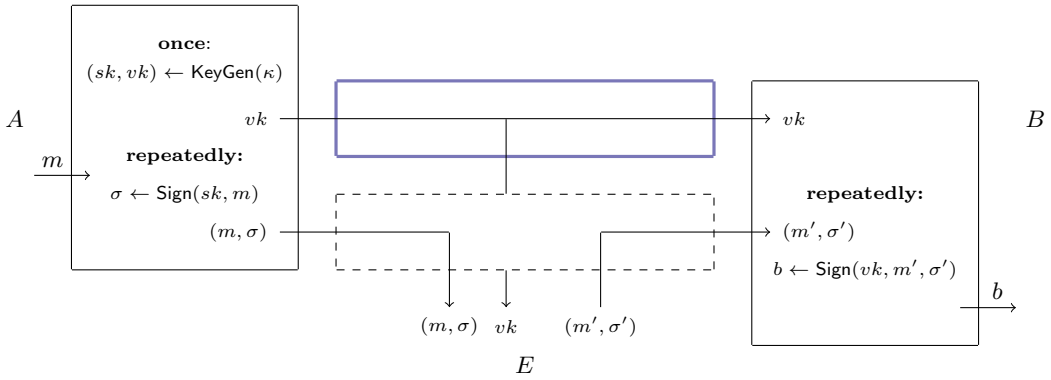


Fig. 11. A signature system, constructed from a one-time authenticated channel and an insecure channel

Remark 9. We have not included the usual kill-switch functionality of the ideal authenticated channel. This is because we need to show that the system realises a multi-use authenticated channel, which does not have such a functionality

(and would thus be complicated to deal with when constructing a simulator). Moreover, one can still implicitly force no message output at Bob's interface simply by inputting no messages into Eve's input port.

The fact that this system realises a multi-use authenticated channel is a simplified (i.e., with two honest communication partners) version of [MTC13, Lemma 3].

Theorem 8 ([MTC13, Lemma 3]). *Let Σ be a signature scheme and $S(\kappa)$ be the associated signature system. For any computationally bounded distinguisher, there exists a bounded adversary $\mathcal{A}$ such that the signature system $\epsilon(\kappa)$-realises an ideal multi-use authenticated channel, where $\epsilon(\kappa) = \mathsf{Adv}^{\mathsf{euf-cma}}_{\Sigma,\mathcal{A}}(\kappa)$. In the language of the cryptographic algebra,*

$$(\pi^{\Sigma}_{\mathsf{sgn}})^{A}(\pi^{\Sigma}_{\mathsf{vfy}})^{B}\left((\mathcal{C}^{*})^{A\to B}||\mathsf{ot}-\mathsf{ac}^{A\to B}\right) \overset{\epsilon(\kappa)}{\longmapsto} \mathsf{m}-\mathsf{ac}^{A\to B}$$

Proof. (*Sketch.*) The idea is very similar to the argument we gave in Theorem 6: the reduction equips $S(\kappa)$ with an MBO such that the combined system's behaviour is equivalent to the signature security game, whereupon we can apply Theorem 1.

Necessity of Assuming Authentication. Compared to the symmetric key setting, we have assumed weaker resources in the construction of an ideal, multi-use authenticated channel: we assume only a one-time authenticated channel, rather than an ideal key-sharing resource (which can be thought of as an ideal, two way, private and authentic channel that shares exactly one value). Since the adversary is now allowed to view the public key, we might ask if we can relax these assumptions further. In particular, we might wonder if it is possible to also distribute the verification key with an insecure channel. The following shows that this is not so: if we try to construct our system from only an insecure channel, a distinguisher has a method of distinguishing the real system from an ideal resource with probability one.

Suppose the verification key is instead sent down an insecure channel, and read by the distinguisher at the adversarial interface. The distinguisher can run KeyGen to generate a key pair (vk_D, sk_D), and deliver vk_D instead of vk_A. It can then choose some message m' that it never input at A's interface, compute $\sigma' = \mathsf{Sign}(sk_D, m')$, and input (m', σ') at the adversarial interface. By correctness of the signature, the real system will always output m' at B's interface, whereas, since m' was never stored in the ideal resource's internal buffer, the ideal resource will always output $\perp$ at B's interface. Such a distinguisher therefore has probability 1 of correctly deciding which system it is interacting with.

Advanced Functionalities There are various generalisations of the notion of a signature to a multi-party context, including blind signatures [BK25], threshold signatures [SY23], and ring signatures [BKM06]. Much of the modern literature on these topics is concerned with realising such functionalities from post-quantum assumptions.

Coming up with standard security notions is not by itself a trivial task[7], and therefore defining a ideal functionality is also somewhat involved. Composable security analyses in the UC framework do exist: for threshold signatures in [Duf+25; HCL08; AF04; HL13]; for ring signatures in [YO07; BDW22; WZL24]; and for blind signatures in [AO12; Fis06; DMO06], though these works should be understood as a small sample of those dealing with the composability of a small sample of advanced signature functionalities.

We are not aware of any work in the CC framework, and any such work would run into the multi-party issues with CC that we will discuss later in Section 5. As such, we regard this as out of scope for this review.

Public Verifiability Digital signatures have a property that appears to be unique among the classical methods of authentication: *public verifiability*. Because the public key is used for verification, a distinguisher can verify the validity of any message-signature pair produced with the corresponding signing key, regardless of whether it interacts with the real or ideal system.

Examples All digital signatures are based on some kind of computational assumption, of which there are 3 types. One should keep in mind that this paper is not a specialist report on digital signatures, nor is it a specialist report on post-quantum cryptography, or the outcome of the NIST standardisation process. Many reviews of this kind exist for those with a deeper interest in the subject.

Classical Assumptions The two most important assumptions here are the difficulty of factoring, and that of the discrete logarithm problem. These problems underpin RSA signatures [RSA78] and Schnorr signatures[8] [Sch89], respectively.

Post-quantum Assumptions In light of the theoretical quantum vulnerability of the above assumptions, various proposals of alternative computational assumptions from which to design digital signatures have been made. These assumptions are advertised as resistant to fast quantum cryptanalysis, and indeed no significant quantum algorithms have been found for them despite extensive research on the topic.

The most important school of such assumptions are those based on the difficulty of solving a wide class of problems in integer lattices. These assumptions underpin the recent NIST standard, ML-DSA [NIS+24]. This standard is effectively the mature version of the CRYSTALS-Dilithium protocol [Duc+18], which can understood to be the 'winner' of the NIST competition for signatures. Another lattice-based signature scheme, Falcon [Fou+18], is slated for

[7] For example, the NIST call for threshold signatures citethreshold-call (in contrast to the flagship post-quantum standardisation call) does not specify a security property that submissions should achieve, and instead merely requires that authors state the type of unforgeability achieved by their submission.

[8] Note that an earlier attempt at providing discrete-log based signatures was made in [DH22], though only partial solutions were provided.

future standardisation, since it offers efficiency trade-offs that may be desirable in certain use cases.

Concerned by the lack of signature schemes within the NIST process based on alternative assumptions (recall that there is evidence for a lack of fast lattice algorithms, but no proof), NIST announced an 'on-ramp' for alternative candidates [SN22]. 14 of these advanced to a second round, and are based, with the successful candidates drawing from problems in coding theory, solving systems of multivariate polynomials, isogenies between elliptic curves, and one-way functions.

It is worth noting that a signature based on the difficulty of solving multivariate polynomial equations progressed into round 3 of the original NIST competition, and could well have been standardised, but succumbed to classical cryptanalysis [Beu22] – that is, the unexpected failure of a post-quantum algorithm in which the community placed a high degree of confidence is not unprecedented[9].

One-way Assumptions It is also possible to construct digital signatures based on the security of one-way functions, based on an idea of Lamport [Lam79]. In some sense, assuming the computational intractability of inverting such a function is a more reasonable assumption than assuming the existence of trap-door functions. FIPS 205, another one of the three extant post-quantum standards, is such a signature.

4.2 KEM Authentication

There has been some work recently [SSW20; Bat+24; Boy+09; Fuj+15], particularly in the context of quantum-safe communication, on establishing authenticated channels with the use of a key encapsulation mechanism (KEM). Let us introduce these objects.

Definition 15. *A* KEM Ψ *is a trio of algorithms* KeyGen, Enc, Dec. *Associated to these algorithms are a an asymmetric key space* $\mathcal{K}_s \times \mathcal{K}_p$*, a symmetric key space* $\mathcal{K}$*, and an encapsulation space* $\mathcal{C}$*, consisting of all the admissible private-public key pairs, symmetric keys, and encapsulations, respectively. The algorithms are defined as follows:*

- KeyGen *takes a security parameter* κ *as an input, and outputs a key pair* (sk, pk) *(sometimes called the 'secret key' and the 'public key', respectively)*
- Enc *takes as input a public key* $pk \in \mathcal{K}_p$*, and outputs a symmetric key* τ *sampled uniformly at random from* $\mathcal{K}$*, and an encapsulation* $z \in \mathcal{C}$*.*
- Dec *takes as input a secret key* $sk \in \mathcal{K}_s$ *and an encapsulation* z*, and outputs either* $\perp$ *(error), or a symmetric key* τ*.*

[9] In fact, it happened twice in the same year. We will discuss the other one when we give examples of KEMs.

The KEM is said to be correct if for any key pair (sk, pk) obtained from KeyGen, if $(z, \tau) = \mathsf{Enc}(pk)$, then $\mathsf{Dec}(sk, z) = \tau$.

A few remarks are in order. First, note that the possible $\perp$ output is now called 'error', rather than 'reject'. This is to reflect that in general, a small probability of decapsulation failure is allowed (with particular relevance for the new post-quantum standards). We have not insisted that the input z to the algorithm Dec be contained in $\mathcal{C}$, essentially since we would like for adversaries to query strings without knowledge of whether they are valid ciphertexts. Nevertheless, Dec should always output $\perp$ if run on an input $c \notin \mathcal{C}$. Finally, we note that a KEM scheme is the dual of a digital signature, in the sense that for a digital signature, the sender holds the key that must be kept secret; the reverse is true for a PKE scheme.

KEMs are closely related to the notion of public-key encryption. Intuitively, a KEM can be understood as the result of generating a random value for use as a key, then encrypting this random value with respect to a PKE scheme. On the other hand, not all KEMs arise in this way, and so a separate treatment is given here for the sake of clarity.

Security of KEMs The security of a KEM is defined in terms of a different type of game to the ones we have seen so far. Instead of trying to construct its own valid cryptography object, the adversary will now be asked to determine which of two possible scenarios it is in.

Since pk is assumed to be public, the adversary trivially has access to ciphertexts computed on messages of its choice. In other words, all KEMs must have so-called cpa (chosen-plaintext attack) security (this is not the case in the symmetric key variant of encryption). An adversary does not automatically have access to messages corresponding to ciphertexts of its course, however. A KEM resistant to an adversary endowed with such an oracle is said to have cca (chosen-ciphertext attack) security; in fact we will need this stronger notion, and so it is the one we put forward.

Definition 16. *Let $\Psi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a KEM, and let $\mathsf{Exp}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa)$ be the experiment described in Figure 12. The advantage of $\mathcal{A}$ in the $\mathsf{ind-cca}$-indistinguishability game against Ψ, denoted $\mathsf{Adv}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa)$, is a function of the security parameter defined by*

$$\mathsf{Adv}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa) = \mathbb{P}\left(\mathsf{Exp}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa) = 1\right)$$

The idea is that the encapsulation should reveal no information about the associated symmetric key, in that it is difficult to distinguish whether the real symmetric key, or a random value, was given. Notice that we have to prohibit querying the challenge encapsulation $\tilde{z}$ to the Dec oracle to prevent a trivial winning strategy.

Composable Security We must first define the ideal resource we wish to prove the security of. Unlike the other cryptographic mechanisms we have discussed,

Algorithm 8 Game $\mathsf{Exp}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa)$

$(pk, sk) \leftarrow \mathsf{KeyGen}(\kappa)$
$(\tilde{z}, \tilde{\tau_0}) \leftarrow \mathsf{Enc}(pk)$
$\tilde{\tau_1} \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{Dec}()}(\tilde{z}, \tilde{\tau}_b, pk)$
return $b' \stackrel{?}{=} b$

Algorithm 9 Oracle $\mathsf{Dec}(z)$

if $z \neq \tilde{z}$ **then**
  return $\mathsf{Dec}(sk, z)$
else
  return $\perp$
end if

Fig. 12. The KEM indistinguishability game in the presence of chosen-ciphertext attacks.

KEMs have a privacy guarantee, and so we will realise something different to the ideal authenticated channels we have discussed so far. In fact, we will realise something called a *ideal unilateral key resource* - in order to gain some intuition on its definition, let us discuss how KEMs can be used for authentication.

Roughly speaking, if one receives a public key pk from an authenticated channel and computes $(z, \tau) = \mathsf{Enc}(pk)$, then only parties in possession of the corresponding secret key can read the symmetric key τ, up to the security of the KEM. This is because the authenticity of the public key received ensures that the public key is bound to the appropriate identity. On the other hand, an adversary is free to inject their own encapsulation against the public key, in which case the symmetric key the intended sender will use is not known to the adversary, but is also not shared with the intended receiver[10].

An important, but rather subtle, distinction now arises. Under the above protocol, the party sending an encapsulation always receives a symmetric key, regardless of whether it is shared by the intended receiver. This corresponds to the notion of *implicit* authentication. We may, however, wish for the sender only to receive a symmetric key when it is shared by the receiver - this corresponds to *explicit* authentication.

Unilateral key resources have been studied in two works in the constructive framework. In fact, the ideal resource put forward in [MTC13] is a unilateral key resource with explicit authentication, whereas the one put forward in [CMT13] is a unilateral key resource with implicit authentication.

In Figure 13, we detail both of these resources. The adversary is allowed to inject their own value to be outputted at B's interface if they set $r = 1$; if they set $r = 0$, μ is output at both interfaces and E has no information on μ. In the implicit case, A always gets τ; in the explicit case, A will never get μ if B gets $\tilde{\mu}$, but it is possible that an adversary 'cuts' the channel in such a way that B gets μ, but A does not. To reflect this capability the adversary gains an additional

[10] Owing to the deployment in practice of this kind of security guarantee, it is customary in the literature to refer to the parties we are calling 'sender' and 'receiver' as the 'client' and 'server', respectively.

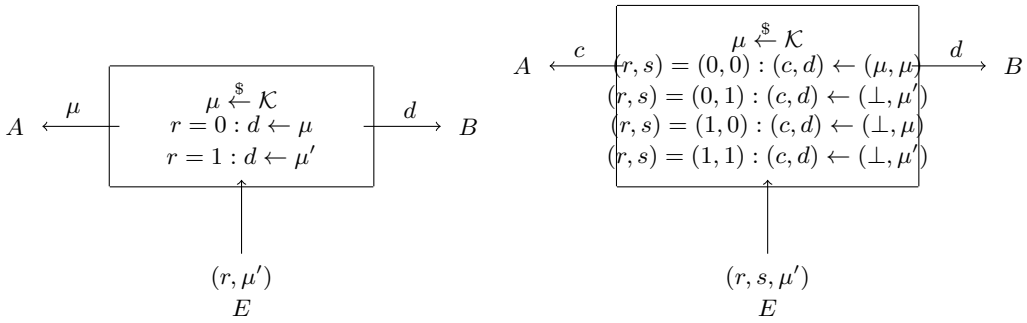


Fig. 13. *Left*: the ideal implicitly unilateral key resource. *Right*: the ideal explicitly unilateral key resource.

input variable s, which can be thought of as a variable encoding whether or not A is allowed to receive a key[11].

The security guarantee of this type of resource is that A can be sure that any key it receives is shared with at most one party, and that party is its intended communication partner. The stronger, explicit version guarantees that any key A receives is shared with its intended communication partner. Note that both flavours are strictly weaker than the ideal key sharing resource, where B can also trust that a key it receives is shared with A. Such a resource is called *bilateral* (as opposed to unilateral).

It is shown in [MTC13] that one can construct an explicitly unilateral key resource from a one-time authenticated channel, a signature scheme, and a cpa-secure KEM. However, the KEM plays no part in achieving authenticity in this case, so we omit further discussion of these results (one can see them as a composable proof of a TLS-like scheme).

More pertinently for us, in [CMT13] it is shown that one can construct an implicitly unilateral key resource from a one-time authenticated channel, an insecure channel, and an $\mathsf{ind-cca}$ secure KEM. Indeed, we have the following:

Theorem 9 ([CMT13, Theorem 2]). *Let Ψ be a KEM, and consider the following converters attached to $(\mathsf{ot-ac}^{B\to A}||(\mathcal{C}^*)^{A\to B})$: $\pi^{\mathsf{KEM}}_{\mathsf{rcv}}$ runs KeyGen internally, and outputs pk at its inside sub-interface corresponding to $\mathsf{ot-ac}^{B\to A}$. When it receives some z at its inside sub-interface corresponding to $(\mathcal{C}^*)^{A\to B})$, it runs $\mathsf{Dec}(sk, z)$ and outputs the resulting τ at its outer interface. $\pi^{\mathsf{KEM}}_{\mathsf{snd}}$ takes as input a public key at its inside sub-interface corresponding to $\mathsf{ot-ac}^{B\to A}$, and runs Enc on this public key to obtain a pair (z, τ). It outputs τ at its outside interface, and inputs z at its inside sub-interface corresponding to $(\mathcal{C}^*)^{A\to B}$. Let $\mathsf{UKR}^{B\to A}_{\mathsf{impl}}$ be the implicitly unilateral key resource. For any computationally bounded distinguisher there exists a computationally bounded adversary $\mathcal{A}$ such*

[11] We include this in our modelling for continuity – this is how the resource is described in [MTC13]. As it happens, a later proof will not work without this functionality.

that

$$(\pi^{\mathsf{KEM}}_{\mathsf{snd}})^A(\pi^{\mathsf{KEM}}_{\mathsf{rcv}})^B(\mathsf{ot}-\mathsf{ac}^{B\leftarrow A}||(\mathcal{C}^*)^{A\to B}) \overset{\epsilon(\kappa)}{\longmapsto} \mathsf{UKR}^{B\to A}_{\mathsf{impl}}$$

where $\epsilon(\kappa) = \mathsf{Adv}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa)$.

Proof. (Sketch). The proof works via the first type of reduction we saw in Section 2.1.One can think of the reduction in this case as a wrapper around the security game that emulates the relevant system to the distinguisher, such that in the two cases of the KEM game the resulting system is distributed either exactly as the real system or the ideal system. Since the adversary can inject its own choice of ciphertext at the insecure channel interface with respect to the real public key, it can see decapsulations of ciphertexts of its choice, hence the need for the $\mathsf{ind}-\mathsf{cca}$ security.

This result is in line with the results of [SSW20], which also requires the $\mathsf{ind}-\mathsf{cca}$ security of the KEM used for authentication. In fact, in that paper it is pointed out that only a single decapsulation query is required, which they term $\mathsf{ind}-\mathsf{1cca}$ security[12].

From Implicitly Unilateral Keys to One-Time Authenticated Channels As demonstrated in [ABD26], one can construct a one-time authenticated channel, from B to A, directly from a implicitly unilateral key resource. Let $\mathcal{H} : \mathcal{M} \to \mathcal{T}$ be a family of strongly universal hash functions. The real system is defined as in Definition 4, except the converter that computes tags is attached to Bob's interface, and the converter that verifies tags is attached to Alice's interface. Note that the adversary now has an extra interface corresponding to the implicitly unilateral key resource. The system is depicted in Figure 14. We describe the system formally in the language of the cryptographic algebra, and prove its security.

Theorem 10. *We have that, for any unbounded distinguisher,*

$$(\pi^{\mathsf{s-auth}}_{\mathsf{vfy}})^A(\pi^{\mathsf{s-auth}}_{\mathsf{tag}})^B(\mathsf{iUK}^{A\leftarrow B}||(\mathcal{C}^*)^{B\to A}) \overset{\epsilon(\kappa)}{\longmapsto} \mathsf{ot}-\mathsf{ac}^{B\to A}$$

where $\epsilon = 1/|\mathcal{T}|$*-constructs a one-time authenticated channel from* B *to* A.

Proof. (Sketch). The idea is similar to Portmann's proof of Theorem 2 argument, except we also have to consider the case where the adversary does inject a key, and when she does not. The insight is that when she injects a key, she gains no extra adversarial power, because she could have generated any corresponding tags herself.

Remark 10. A variant of this argument in a multi-party setting, which was developed concurrently with this work, appears as [ABD26, Lemma 2].

The sequential composition theorem immediately gives us the following.

[12] Not to be confused with $\mathsf{ind}-\mathsf{cca1}$ security.

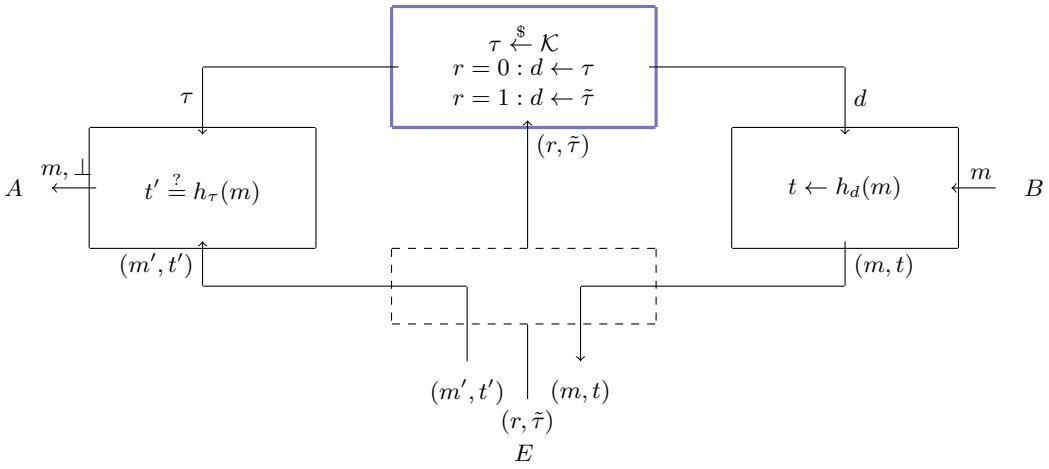


Fig. 14. Realising a one-time authenticated channel from an implicitly unilateral key resource.

.

Corollary 3. *Let Ψ be a* KEM *and $\mathcal{H} : \mathcal{M} \to \mathcal{T}$ be a family of universal hash functions. There exist converters π, π' such that for all computationally bounded distinguishers,*

$$(\pi)^A(\pi')^B(\mathsf{ot} - \mathsf{ac}^{B \to A} || (\mathcal{C}^*)^{A \to B}) \overset{\epsilon(\kappa) + \frac{1}{|\mathcal{T}|}}{\longmapsto} \mathsf{ot} - \mathsf{ac}^{B \to A}$$

where $\epsilon(\kappa) = \mathsf{Adv}^{\mathsf{ind-cca}}_{\Psi, \mathcal{A}}(\kappa)$.

Remark 11. Notice that even though one can construct a one-time authenticated channel from an implicitly unilateral key resource in the presence of an unbounded distinguisher, the security guarantees of the previous result only hold in the presence of a computationally bounded distinguisher. Since either of the two components 'breaking' voids the security guarantees, security in composition only survives in the presence of the weakest type of distinguisher considered.

Remark 12. It is arguably rather circular to construct a one-time authenticated channel from a one-time authenticated channel in this way. The motivation for demonstrating such a result are somewhat out of scope, but have to do with forward secrecy in an authenticated key-exchange context. Moreover, the one-time authenticated key exchange assumed in the setup is essentially a stand-in for the idea of a public-key infrastructure, which we will discuss further later on.

From Implicitly Unilateral Keys to Multi-use Authenticated Channels It is a natural question to ask whether we can also construct multi-use authenticated channels using KEMs. We informally summarise three methods of doing so – the ideas underpinning full security proofs are very similar to those we have already seen, and so those are omitted.

There are three methods by which we could construct a multi-use authenticated channel from a implicitly unilateral key resource. First, we could simply construct a MAC system that uses an implicitly unilateral key resource instead of an ideal key distribution resource. Similarly to our argument in the proof of Theorem 10, if the distinguisher does not inject a key the system is distributed identically to the system using an ideal key distribution resource; if the distinguisher injects a key, it effectively loses its cma power (because it could have computed the tags it sees itself, having chosen the key), and so its advantage does not exceed the case in which no key was injected. One could further conclude, similarly to Corollary 3, that a KEM and a MAC construct a multi-use authenticated channel in the presence of a computationally bounded distinguisher; indeed, we have the following:

Corollary 4. *Let Ψ be a KEM, and MAC be a message authentication code. There exist converters π, π' such that for all computationally bounded distinguishers,*

$$(\pi)^A(\pi')^B \left(\mathsf{ot} - \mathsf{ac}^{B\to A} || (\mathcal{C}^*)^{A\to B}\right) \overset{\epsilon(\kappa)+\epsilon'(\kappa)}{\longmapsto} \mathsf{m} - \mathsf{ac}^{B\to A}$$

where $\epsilon(\kappa) = \mathsf{Adv}^{\mathsf{ind-cca}}_{\Psi,\mathcal{A}}(\kappa)$, and $\epsilon'(\kappa) = \mathsf{Adv}^{s-uf}_{\mathsf{MAC},\mathcal{A}}(\kappa)$.

Now that we know how to construct a one-time authenticated channel from an implicitly unilateral key resource, we can apply the idea of [MTC13, Lemma 4] to construct an explicitly unilateral key over this authenticated channel by using an $\mathsf{ind}-\mathsf{cpa}$ secure KEM. We can then use a MAC to construct a multi-use authenticated channel. The benefit of this approach compared to the prior one is forward security; if the internal secret of the implicitly unilateral key resource ever leaks once the explicitly unilateral key resource has been constructed, the symmetric secret used for authentication is independent of this secret, and the security survives. There is not currently a way to formalise these types of forward secrecy arguments in the constructive cryptography framework.

Finally, [CMT13] points out that one can establish a unilateral key resource that gives out n keys, at the cost of inducing a factor of n in the advantage bound. We could use each of these keys with universal hash functions to establish a multi-use authenticated channel as multiple concurrent single-use channels. This is probably the weakest option: using a KEM is significantly more computationally expensive than using a MAC, and we still only have security in the presence of computationally bounded distinguishers. We also do not receive any forward security guarantees.

Remark 13. In this section we have used a technique of defining an intermediary ideal functionality and applying the composition theorem. This can be thought of as an analogous to the usual game-hopping technique, can also be found in the constructive analysis of TLS 1.3 [Koh+15].

Examples The following section is very similar to our brief overview of digital signatures (see Section 4.1), and should be understood to be subject to the same

disclaimers. Notice that it is not known to be possible to construct KEMs from one-way functions, and so we have only two categories of assumption from which these schemes are constructed.

Classical Assumptions The most relevant examples here are public-key encryption schemes, originally intended to encrypt messages directly. These are the famous RSA encryption scheme of [RSA78], based on factoring, and the encryption scheme of Elgamal [ElG85], based on the discrete logarithm problem. Notice that it is not the modern practice to achieve privacy with a KEM; in fact, TLS 1.3 establishes ephemeral secrets with (a modern version of) the discrete-log based Diffie-Hellman key exchange protocol of [DH22], essentially for efficiency reasons. This latter protocol is an example of a non-interactive key exchange, of which there are few post-quantum candidates (with the notable exception of CSIDH and its variants [Cas+18]).

Post-quantum Assumptions Again, the most important class of post-quantum cryptography proposals are those based on lattice problems, which also underpin the KEM standard, ML-KEM. This standard is based on CRYSTALS-Kyber [Bos+18]. Another KEM, HQC [Mel+18], is now also slated for standardisation. This protocol is based on the difficulty of a certain problem in linear coding theory.

We also record the other high-profile example of a loss of confidence in a post-quantum cryptography candidate: the KEM known as SIDH [JD11], based on the difficulty of providing a certain type of map between classes of elliptic curve, succumbed to classical cryptanalysis at an advanced stage of the NIST competition [CD23].

5 The Multi-party Setting

So far we have constructed authenticity in the setting whereby there are exactly two honest parties, and a single adversarial party. This is sometimes called the $\{A, B, E\}$ setting. In this instance, each honest party has only a single possible honest communication partner; if a cryptographic mechanism accepts, then either the communication originated from the intended partner, or the adversary was successful in breaking the cryptography.

The usual model for communication between multiple honest parties might loosely be termed the 'authenticated key exchange' setting. Pioneered by Bellare and Rogaway in a series of papers in the 1990s, in this setting all communication between honest parties is controlled by a single, powerful adversary that can copy, drop, re-route and delay honest communication. The security requirement then becomes that the adversary can do no better than to faithfully transmit communication.

5.1 Parallel Composition

Moving to a multi-party setting requires more care than in the $\{A, B, E\}$ case, because now the adversary can choose which of multiple partners receive a par-

ticular message. A particular run of a protocol is called a *session*. We require that both participants in a session agree on the identities of the participants, and how many times they have already run a session between them – these requirements correspond to resistance to identity misbinding attacks, and replay attacks, respectively. More modern protocols (such as TLS 1.3 and KEM-TLS) further subdivide sessions into *stages*, and enforce further matching guarantees with respect to these stages.

Ideally, in the spirit of the modularity of constructive cryptography, we would analyse protocols in the $\{A, B, E\}$ setting, and apply parallel composition to move to the multi-party case. Let us see the effectiveness of this approach in various settings.

Symmetric Key Authentication In this case any two parties require a distinct key between them, and each party has to maintain as many keys as it has communication partners. As such, the keys themselves can act as identifiers, and there is little difference between the 2-party and multi-party case; that is, applying parallel composition is unproblematic.

Signatures In this scenario we run into the problem of shared state between parallel runs of the protocol, and composition becomes trickier. We illustrate this with an example: imagine that there are three honest parties, A_1, A_2 and B, and that A_1 and A_2 have a genuine copy of B's verification key. Suppose B wishes to transmit classical data m to A_1, and so computes a signature on m; but the adversary instead delivers this message-signature pair to A_2. A_2 will verify, but both parties have a different view of who is participating in the communication.

The trouble here is that the parallel composition of ideal resources and real resources are not the same, which one can see as follows. The ideal system is the composition of two multi-use authenticated channels; the simulator at some point outputs message-signature pairs (m_1, σ_1), (m_2, σ_2), and receives from the distinguisher $(m_1', \sigma_1'), (m_2', \sigma_2')$. If, for example, the distinguisher sets $m_1' = m_2, \sigma_1' = \sigma_2$, the simulator will pass m_2 to the first ideal authenticated channel – which will output reject, since m_2 was never input into that resource's internal buffer. As we have seen in the paragraph above, the real system will not reject, so the distinguisher has a high probability strategy for distinguishing these two resources.

What has gone wrong? The answer is that there is a shared state between the parallel instances of the real system - namely, the key pair. The parallel composition theorem does not automatically cater for such a scenario. In [MTC13] the issue is averted by arguing directly about the composition of resources, rather than relying on the composition theorem to act as a compiler, and introducing a 'naming' resource – effectively, locally generated random strings, which the receiving parties will only verify with respect to. It suffices to distribute these over an insecure channel, so this resource is in some sense 'free'.

This is still not completely satisfactory: it suffices to prove the security of a given network, but the composition of this network with another instance of the

real system (that is, wishing to send data to another party). Indeed this is particularly troublesome given that the main advantage of public-key authentication is in its scalability.

A similar issue applies for the composition of KEMs, as we will now see.

KEMs This time, consider constructing a unilateral key resource with a KEM from a one-time/one-message authenticated channel, as in Theorem 9. Imagine the distinguisher swaps the ciphertexts and delivers them. In the ideal case, the simulator, which does not have access to the unilateral key resource's internal secrets, injects its own choice of secret and causes those to be output. In the real case, the real secrets are simply swapped. Again, the distinguisher has a high probability strategy for distinguishing these two resources.

Getting around this technicality is an open question. We note, however, that both the signature and KEM case could potentially be solved by a more general version of a parallel composition theorem that allowed for shared state across parallel protocol instances.

A somewhat 'brute-force' solution is offered in [ABD26], in which a single-server-many-client variant of the ideal resource is analysed. This necessitates a great deal of syntactical bookkeeping which is somehow not in the spirit of the modularity of constructive cryptography; it would be much cleaner to be able to compile the $\{A, B, E\}$ setting directly.

5.2 Public Key Infrastructure

Thus far we have seen that public-key authentication can be constructed from an ideal one-time authenticated channel. Theoretically, this is a strictly weaker requirement than the ideal key resources we required for symmetric authentication. In practice, however, it is not immediately clear that establishing a one-time authenticated channel should be easier than sharing a one-time secret ideally; for example, an in-person meeting would suffice for both tasks. Moreover, each time one wishes to communicate authentically with a new partner, a fresh and distinct ideal one-time authenticated channel must be established with that partner. It seems that we have incurred weaker security guarantees, without obviously improving scalability.

The reason this is not so is because of the transferable verification of digital signatures; that is, the publicity of the verification key means anyone can verify a message-signature pair generated with the corresponding signing key. This transferability allows us to delegate authenticity to a trusted third party, in such a way that we only ever require one call to an ideal, one-time authenticated channel, no matter how many partners we wish to communicate with. Although the term has no strict definition, a mechanism by which one can realise this delegation of authentication is called a *Public Key Infrastructure* (PKI), which will typically involve a *Certificate Authority* (CA).

We discuss a simplified version of a PKI that works roughly as follows: a network consists of many users, and a single, mutually trusted authority (the

CA). Using a one-time, two-way authenticated channel, the user can submit their public key to the certificate authority, and receive the CA's public key, along with a signature of the user's public key computed with respect to the CA's signing key. Any user can now transmit their public key authentically by including the CA's signature, provided the recipient has also authentically received the CA's public key. A packet of data containing the sender's public key, plus the CA's attestation of its validity, is called a *certificate*. Another way of thinking about this is that the certificate provides a means of establishing a one-message authenticated channel Crucially, the required authentic communication with the CA is one-time, regardless of new users entering the network. Another way of thinking about this is that the CA has provided us with a means of establishing a one-message authenticated channel with any user who has also established a round of authentic communication with the CA; we only received a signature on our public key, so only this particular string can be successfully verified.

An important note is that the authenticated communication with the CA is one-time and enables communication with an unlimited number of users who have also established one-time authenticated communication with that CA.

Relationship with Key-Distribution Centres A similar idea is possible in the symmetric key case. This time, users establish a long-term symmetric key with the trusted authority via a one-time appeal to an ideal key distribution resource. Every time they wish to communicate privately and authentically with another user, the trusted authority can mediate the establishment of an authentic session key, provided that both users share a symmetric key with the trusted authority. However, the trusted authority has to be online to facilitate this. Our thesis is that public key cryptography can be understood as an *offline key-distribution centre*, where the flexibility implied by not requiring a trusted authority to be online comes at the cost of incurring additional computational assumptions.

Public-Key Infrastructure in the Composable Literature Public-key infrastructure has primarily been addressed in the UC framework, notably in [CSV16]. This is an application of the Global UC framework, which tries to get around data provided by public-key infrastructure occurring in many parallel sessions. Note that this sort of concern is related to the issues with parallel composition we discussed above. In the CC framework, assumption of a set-up one-time authenticated channel in [MTC13] can be seen as rudimentary modelling of a PKI, although, as we have pointed above, various subtleties remain.

6 Conclusion

We have completely mapped the landscape of constructing composable authentication in the $\{A, B, E\}$ setting, systematising existing results and filling in

various gaps in the literature. The remaining gaps concern more advanced functionalities, particularly in a multi-party regime. As we have highlighted, a more precise constructive modelling of public key infrastructure would be particularly welcome.

Recall that we have highlighted the shape of the literature in Figure 1.

References

[ABD26] Myrto Arapinis, Christopher Battarbee, and Mina Doosti. "Have Your CKAKE and Eat it, Too: Efficient, Composable KEM-Authenticated Key Exchange". In: *Cryptology ePrint Archive* (2026).

[AF04] Masayuki Abe and Serge Fehr. "Adaptively secure Feldman VSS and applications to universally-composable threshold cryptography". In: *Annual International Cryptology Conference*. Springer. 2004, pp. 317–334.

[AO12] Masayuki Abe and Miyako Ohkubo. "A framework for universally composable non-committing blind signatures". In: *International Journal of Applied Cryptography* 2.3 (2012), pp. 229–249.

[Bat+24] Christopher Battarbee, Christoph Striecks, Ludovic Perret, Sebastian Ramacher, and Kevin Verhaeghe. "Quantum-safe hybrid key exchanges with kem-based authentication". In: *arXiv preprint arXiv:2411.04030* (2024).

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. "Public-key encryption in a multi-user setting: Security proofs and improvements". In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2000, pp. 259–274.

[BDW22] Pedro Branco, Nico Döttling, and Stella Wohnig. "Universal ring signatures in the standard model". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2022, pp. 249–278.

[Bel+97] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. "A concrete security treatment of symmetric encryption". In: *Proceedings 38th annual symposium on foundations of computer science*. IEEE. 1997, pp. 394–403.

[Beu22] Ward Beullens. "Breaking rainbow takes a weekend on a laptop". In: *Annual International Cryptology Conference*. Springer. 2022, pp. 464–479.

[BGM04] Mihir Bellare, Oded Goldreich, and Anton Mityagin. "The power of verification queries in message authentication and authenticated encryption". In: *Cryptology ePrint Archive* (2004).

[BK25] Aditya Bhardwaj and Péter Kutas. "A Gentle Introduction to Blind signatures: From RSA to Lattice-based Cryptography". In: *arXiv preprint arXiv:2509.02189* (2025).

[BKM06] Adam Bender, Jonathan Katz, and Ruggero Morselli. "Ring signatures: Stronger definitions, and constructions without random oracles". In: *Theory of Cryptography Conference*. Springer. 2006, pp. 60–79.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. "The security of the cipher block chaining message authentication code". In: *Journal of Computer and System Sciences* 61.3 (2000), pp. 362–399.

[Bla+99] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. "UMAC: Fast and secure message authentication". In: *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer. 1999, pp. 216–233.

[Bos+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM". In: *2018 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2018, pp. 353–367.

[Boy+09] Colin Boyd, Yvonne Cliff, Juan M Gonzalez Nieto, and Kenneth G Paterson. "One-round key exchange in the standard model". In: *International Journal of Applied Cryptography* 1.3 (2009), pp. 181–199.

[Can04] Ran Canetti. "Universally composable signature, certification, and authentication". In: *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.* IEEE. 2004, pp. 219–233.

[Cas+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International conference on the theory and application of cryptology and information security*. Springer. 2018, pp. 395–427.

[CD23] Wouter Castryck and Thomas Decru. "An efficient key recovery attack on SIDH". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2023, pp. 423–447.

[CMT13] Sandro Coretti, Ueli Maurer, and Björn Tackmann. "A constructive perspective on key encapsulation". In: *Number Theory and Cryptography: Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday*. Springer, 2013, pp. 226–239.

[CSV16] Ran Canetti, Daniel Shahaf, and Margarita Vald. "Universally composable authentication and key-exchange with global PKI". In: *Public-Key Cryptography–PKC 2016*. Springer, 2016, pp. 265–296.

[CW77] J Lawrence Carter and Mark N Wegman. "Universal classes of hash functions". In: *Proceedings of the ninth annual ACM symposium on Theory of computing*. 1977, pp. 106–112.

[DH22] Whitfield Diffie and Martin E Hellman. "New directions in cryptography". In: *Democratizing cryptography: the work of Whitfield Diffie and Martin Hellman*. 2022, pp. 365–390.

[DMO06] Seiji Doi, Yoshifumi Manabe, and Tatsuaki Okamoto. "Universally composable blind signatures". In: *Proc. SCIS2006, Hiroshima, Japan* 2A1-4 (2006).

[Duc+18] Léo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "Crystals–dilithium: Digital signatures from module lattices". In: (2018).

[Duf+25] Antonín Dufka, Semjon Kravtšenko, Peeter Laud, and Nikita Snetkov. "Trilithium: Efficient and universally composable distributed ML-DSA signing". In: *Cryptology ePrint Archive* (2025).

[ElG85] Taher ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[Fis06] Marc Fischlin. "Round-optimal composable blind signatures in the common reference string model". In: *Annual International Cryptology Conference*. Springer. 2006, pp. 60–77.

[Fou+18] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. "Falcon: Fast-Fourier lattice-based compact signatures over NTRU". In: *Submission to the NIST's post-quantum cryptography standardization process* 36.5 (2018), pp. 1–75.

[Fre+24] Giacomo Fregona, Claudia De Lazzari, Damiano Giani, Fernando Chirici, Francesco Stocco, Edoardo Signorini, Guglielmo Morgari, Tommaso Occhipinti, Alessandro Zavatta, and Davide Bacco. "Authentication Methods for Quantum Key Distribution: Challenges and Perspectives". In: *Toward a Quantum-Safe Communication Infrastructure*. Ed. by R. Steinwandt and A. Xuereb. IOS Press, 2024, pp. 54–66. DOI: 10.3233/NICSP240007.

[Fuj+15] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. "Strongly secure authenticated key exchange from factoring, codes, and lattices". In: *Designs, Codes and Cryptography* 76.3 (2015), pp. 469–504.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. "A digital signature scheme secure against adaptive chosen-message attacks". In: *SIAM Journal on computing* 17.2 (1988), pp. 281–308.

[HCL08] Xuan Hong, Ke-fei Chen, and Yu Long. "Universally composable proactive threshold RSA signature". In: *Journal of Shanghai Jiaotong University (Science)* 13.6 (2008), pp. 659–663.

[HL13] Xuan Hong and Luqun Li. "Adaptive Universal Composability Framework for Server-Aided Threshold Signature". In: *Research Journal of Applied Sciences, Engineering and Technology* 5.1 (2013), pp. 270–274.

[IK03] Tetsu Iwata and Kaoru Kurosawa. "Omac: One-key cbc mac". In: *Fast Software Encryption: 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers 10*. Springer. 2003, pp. 129–153.

[JD11] David Jao and Luca De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *International workshop on post-quantum cryptography*. Springer. 2011, pp. 19–34.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[Koh+15] Markulf Kohlweiss, Ueli Maurer, Cristina Onete, Björn Tackmann, and Daniele Venturi. "(De-) constructing TLS 1.3". In: *International Conference on Cryptology in India*. Springer. 2015, pp. 85–102.

[Lam79] Leslie Lamport. "Constructing digital signatures from a one way function". In: (1979).

[Mau10] Ueli Maurer. "Constructive Cryptography – A Primer". In: *Financial Cryptography and Data Security (FC 2010)*. Ed. by Radu Sion. Vol. 6052. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 1–18. DOI: 10.1007/978-3-642-14577-3_1.

[Mau12] Ueli Maurer. "Constructive Cryptography – A New Paradigm for Security Definitions and Proofs". In: *Theory of Security and Applications (TOSCA 2011)*. Ed. by Stefan Mödersheim and Catuscia Palamidessi. Vol. 6993. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 33–56. DOI: 10.1007/978-3-642-27375-9_3.

[Mau13] Ueli Maurer. "Conditional equivalence of random systems and indistinguishability proofs". In: *2013 IEEE International Symposium on Information Theory*. IEEE. 2013, pp. 3150–3154.

[Mel+18] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. "Hamming quasi-cyclic (HQC)". In: *NIST PQC Round* 2.4 (2018), p. 13.

[MR11] Ueli Maurer and Renato Renner. "Abstract Cryptography". In: *Proceedings of the Second Symposium on Innovations in Computer Science (ICS 2011)*. Beijing: Tsinghua University Press, 2011, pp. 1–21.

[MTC13] Ueli Maurer, Björn Tackmann, and Sandro Coretti. "Key exchange with unilateral authentication: Composable security definition and modular protocol design". In: *Cryptology ePrint Archive* (2013).

[NIS+24] NIST, Thinh Dang, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, and Angela Robinson. *Module-Lattice-Based Digital Signature Standard (ML-DSA)*. Federal Information Processing Standard FIPS 204. Gaithersburg, MD: National Institute of Standards and Technology (NIST), 2024. DOI: 10.6028/NIST.FIPS.204. URL: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf.

[Por14a] Christopher Portmann. “Key recycling in authentication”. In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 4383–4396.

[Por14b] Christopher Portmann. “Key recycling in authentication”. In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 4383–4396.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[Sch89] Claus-Peter Schnorr. “Efficient identification and signatures for smart cards”. In: *Conference on the Theory and Application of Cryptology*. Springer. 1989, pp. 239–252.

[SN22] National Institute of Standards and Technology (NIST). *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process*. NIST / CSRC Call for Proposals. Updated October 2022. NIST, Sept. 2022. URL: `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf`.

[SSW20] Peter Schwabe, Douglas Stebila, and Thom Wiggers. “Post-quantum TLS without handshake signatures”. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1461–1480.

[ST08] National Institute of Standards and Technology. *The Keyed Hash Message Authentication Code (HMAC)*. Tech. rep. Federal Information Processing Standards Publications (FIPS) 198-1, Change Notice July 16, 2008. Washington, D.C.: U.S. Department of Commerce, 2008. DOI: `10.6028/NIST.FIPS.198-1`.

[Sti94] Douglas R. Stinson. “Universal hashing and authentication codes”. In: *Designs, Codes and Cryptography* 4.3 (1994), pp. 369–380.

[SY23] Kiarash Sedghighadikolaei and Attila Altay Yavuz. “A comprehensive survey of threshold signatures: Nist standards, post-quantum cryptography, exotic techniques, and real-world applications”. In: *arXiv preprint arXiv:2311.05514* (2023).

[WC81] Mark N Wegman and J Lawrence Carter. “New hash functions and their use in authentication and set equality”. In: *Journal of computer and system sciences* 22.3 (1981), pp. 265–279.

[WZL24] Xingkai Wang, Chunping Zhu, and Zhen Liu. “A universally composable linkable ring signature supporting stealth addresses”. In: *Mathematics* 12.3 (2024), p. 491.

[YO07] Kazuki Yoneyama and Kazuo Ohta. “Ring signatures: Universally composable definitions and constructions”. In: *Proceedings of the 2nd ACM symposium on Information, computer and communications security*. 2007, pp. 374–376.

A Technical Specifications

Here we keep a list of the technical specifications of each of the ideal resources discussed throughout the work, along with a short summary of their functions.

$\mathsf{ot} - \mathsf{ac}^{A \to B}$

Algorithm 10 *

Initialisation

$\mathsf{count} \leftarrow 0$
$d \leftarrow$ “”

Algorithm 11 *

Interface A

1: **input** m
2: $\mathsf{count} \leftarrow \mathsf{count} + 1$
3: **if** $\mathsf{count} == 1$ **then**
4: $\quad d \leftarrow m$
5: **end if**

Algorithm 12 *

Interface B

1: **output** d

Algorithm 13 *

Interface E sub-interface: `read`

1: **output** m

sub-interface `cut`

1: **input** c
2: **if** c **then**
3: $\quad d \leftarrow \perp$
4: **end if**

Fig. 15. The one-time authenticated channel resource, $\mathsf{ot} - \mathsf{ac}^{A \to B}$. The input at A's interface is output at E's interface; E can choose whether or not B receives this input, or an error.

$\mathsf{m} - \mathsf{ac}^{A \to B}$

Algorithm 14 *

Initialisation

$\mathcal{B} \leftarrow \{\}$
$d \leftarrow$ “”

Algorithm 15 *

Interface A

1: **input** m
2: $\mathcal{B} \leftarrow \mathcal{B} \cup \{m\}$

Algorithm 16 *

Interface B

1: **output** d

Algorithm 17 *

Interface E sub-interface: `read`

1: **output** m

sub-interface `inject`

1: **input** m'
2: **if** $m' \in \mathcal{B}$ or $m' ==$ “” **then**
3: $\quad d \leftarrow m'$
4: **end if**

Fig. 16. The multi-use authenticated channel resource, $\mathsf{ot} - \mathsf{ac}^{A \to B}$. The input at A's interface is added to an internal buffer output at E's interface; E can cause any message in the buffer to be output at B's interface in any order. The `inject` sub-interface also accepts the empty message, which is equivalent to interface B outputting an error.

$\mathsf{kd}^{A \leftrightarrow B}$

Algorithm 18 *

Initialisation

$k \xleftarrow{\$} \mathcal{K}$

Algorithm 19 *

Interface A

1: **output** k

Algorithm 20 *

Interface B

1: **output** k

Algorithm 21 *

Interface E

Fig. 17. The ideal key-distribution resource. A key is selected uniformly at random internally, and output at the A and B interfaces. E cannot interact with this resource.

$(\mathcal{C}^*)^{A \to B}$

Algorithm 22 *

Initialisation

Algorithm 23 *

Interface A

1: **input** m

Algorithm 24 *

Interface B

1: **output** m'

Algorithm 25 *

Interface E sub-interface: `read`

1: **output** m

sub-interface `inject`

1: **input** m'

Fig. 18. The insecure channel resource. A's input is given to E, who can freely choose the value output at B's interface.

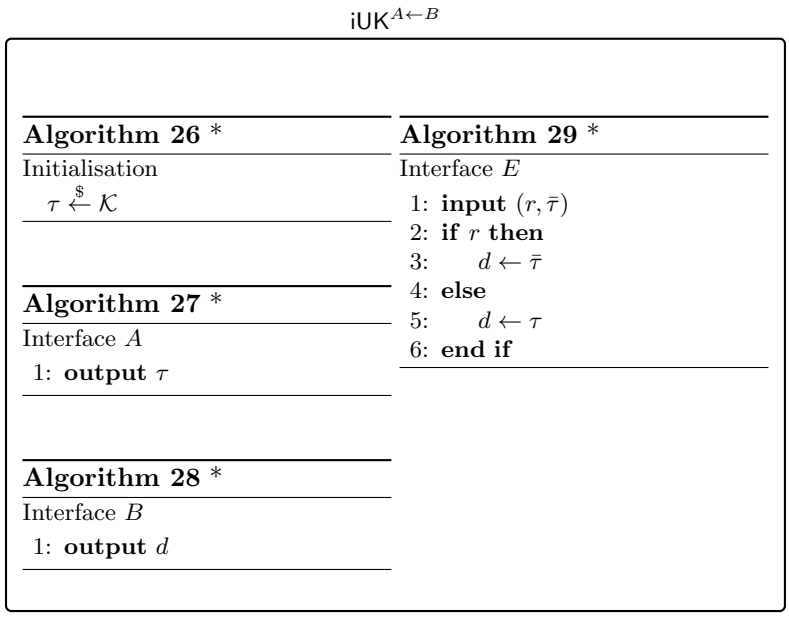


Fig. 19. The implicitly unilateral key resource. A always gets a secret key selected uniformly at random; the adversary can choose whether B also receives this key, or receives a key of its own choosing.

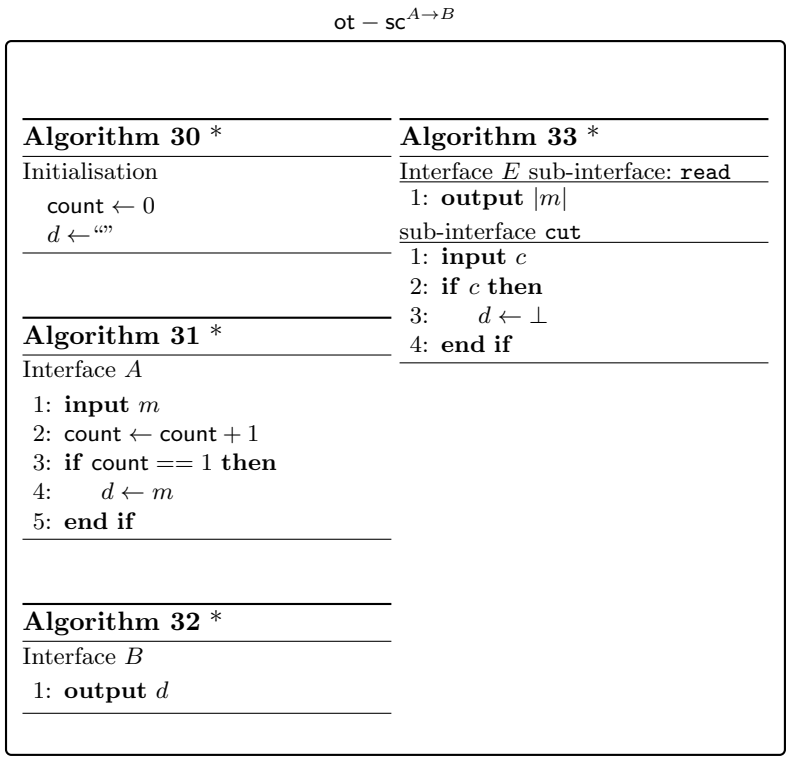


Fig. 20. The one-time secure channel resource, $\mathsf{ot}-\mathsf{sc}^{A\to B}$. The length of the input at A's interface is output at E's interface; E can choose whether or not B receives this input, or an error.

$\mathsf{m} - \mathsf{sc}^{A \to B}$

Algorithm 34 *

Initialisation

$\mathcal{B} \leftarrow \{\}$
$d \leftarrow$ “”
$\mathsf{count} \leftarrow 0$

Algorithm 35 *

Interface A

1: **input** m
2: $\mathsf{count} \leftarrow \mathsf{count} + 1$
3: $\mathcal{B} \leftarrow \mathcal{B} \cup \{i : m\}$

Algorithm 36 *

Interface B

1: **output** d

Algorithm 37 *

Interface E sub-interface: `read`

1: **output** $|m|$

sub-interface `inject`

1: **input** i
2: **if** $i \in \mathcal{B}.\mathsf{keys}()$ **then**
3: $\quad d \leftarrow \mathcal{B}[i]$
4: **else**
5: $\quad d \leftarrow \perp$
6: **end if**

Fig. 21. The multi-use secure channel resource, $\mathsf{m} - \mathsf{sc}^{A \to B}$. The input at A's interface is added to an internal buffer dictionary, and bound to the order in which it was submitted. The length, meanwhile is output at E's interface; E can cause any message in the buffer to be output at B's interface in any order.