Formal Verification of a Post-Quantum Signal Protocol with Tamarin

Hugo Beguinet, Céline Chevalier, Thomas Ricosset, Hugo Senet

Formal Verification of a Post-Quantum Signal Protocol with Tamarin

Hugo Beguinet[1,2], Céline Chevalier[1,3], Thomas Ricosset[2], and Hugo Senet[1,2]

[1] DIENS, École Normale Supérieure, CNRS, Inria, PSL University, Paris, France,
hugo.beguinet, celine.chevalier, hugo.senet@ens.fr
[2] Thales, Gennevilliers, France,
hugo.beguinet, thomas.ricosset@thalesgroup.com,
hugo.senet@thalesgroup.com
[3] CRED, Université Paris-Panthéon-Assas, Paris, France

Abstract. The Signal protocol is used by billions of people for instant messaging in applications such as Facebook Messenger, Google Messages, Signal, Skype, and WhatsApp. However, advances in quantum computing threaten the security of the cornerstone of this protocol: the Diffie-Hellman key exchange. There actually are resistant alternatives, called post-quantum secure, but replacing the Diffie-Hellman key exchange with these new primitives requires a deep revision of the associated security proof. While the security of the current Signal protocol has been extensively studied with hand-written proofs and computer-verified symbolic analyses, its quantum-resistant variants lack symbolic security analyses.
In this work, we present the first symbolic security model for post-quantum variants of the Signal protocol. Our model focuses on the core state machines of the two main sub-protocols of Signal: the X3DH handshake, and the so-called *double ratchet* protocol. Then we show, with an automated proof using the Tamarin prover, that instantiated with the Hashimoto-Katsumata-Kwiatkowski-Prest post-quantum Signal's handshake from PKC'21, and the Alwen-Coretti-Dodis KEM-based double ratchet from EUROCRYPT'19, the resulting post-quantum Signal protocol has equivalent security properties to its current classical counterpart.

Keywords: Secure instant messaging · Signal protocol · Quantum resistant · Formal verification · Tamarin prover · X3DH · Double ratchet.

1 Introduction

The Signal protocol is divided into two sub-protocols: X3DH [21], and the double ratchet protocol [20]. The X3DH protocol can be seen as an Authenticated Key Exchange (AKE) protocol. It ensures the authenticity of an initial key shared between two users. It is an asynchronous protocol, which means that there is no need for users to be online at the same time to initialize the protocol. To use the X3DH protocol, each user must first generate a long-term static pair of public and private keys for them to be authenticated, as well as a batch of ephemeral pairs of public and private keys. Both

long-term public keys and ephemeral key batches are then stored on an honest intermediate server which acts as a buffer. When Bob wants to start a conversation with Alice, he sends a request to the server, and then receives the Alice's long-term public key and a fresh Alice's ephemeral public key from her batch. These two public keys enable Bob to perform the X3DH handshake protocol by sending a message to Alice, which will enables her to derive their X3DH pre-shared secret key when she is online.

The double ratchet protocol is used to encrypt messages to send through the Signal protocol with an Authenticated Encryption with Associated Data (AEAD) scheme and a session key that is shared between the two parties. The session key is renewed each time a message is sent, using symmetric and asymmetric mechanisms called ratchets. The double ratchet protocol is initialized with the X3DH pre-shared key as session key, and an ephemeral public key from the corresponding batch as public key. Then, to send a message, the public key is used to exchange a fresh secret key, from which the new session key is derived with the output of a one-way function applied to the current session key. In addition, a new ephemeral key pair is generated whose public key is encrypted then sent with the message, using this new session key. This protocol is repeated for each message sent to ensure strong security properties such as forward secrecy and post-compromise recovery against passive adversaries.

The current Signal protocol heavily uses the well-known, flexible, and efficient, but vulnerable to quantum attacks, Diffie-Hellman (DH) key exchange protocol. However, with the threat of upcoming quantum computers, post-quantum alternatives are subject to extensive analysis in order to gain assurance in their security. In 2016, the NIST initiated a process to evaluate and standardize quantum-resistant key-establishment and signature schemes, but all remaining candidates in the key-establishment category are key encapsulation mechanisms (KEMs) like RSA, and not key exchanges like DH. Consequently, the integration of post-quantum KEMs in cryptographic protocols is quite challenging due to the differences between KEMs and DH, which requires some fundamental adjustments to these protocols to maintain the same security guarantees.

Aside from that, the active area of formal protocol verification is increasingly accompanying protocol specifications. Designing cryptographic protocols is known to be hard to get right and hand-written proofs remain highly complex and error-prone. At the design level, automatic verification aims to manage the complexity of security proofs and even reveal subtle flaws or as-yet-unknown attacks as the historic example of the man-in-the-middle attack [16]. Efficient automatic verification tools as Tamarin [17] or ProVerif [6] have been used to analyze large, real-world protocols. For instance, ProVerif has been used to analyze TLS 1.3 [4] and Signal [14] and Tamarin as been used to analyze the 5G AKE protocol [3] and TLS 1.3 [10].

Related Works. The security of the (EC)DH-based Signal protocol has been extensively studied using hand-written proofs [9]. Those proofs were completed with a symbolic analysis [14] using the ProVerif prover. Regarding the transition to post-quantum cryptography, there are KEM-based alternatives to the Signal sub-protocols X3DH [12,7] (the security properties in [12] being closer to that of X3DH, in particular thanks to the encryption of the signature) and double ratchet [1], with hand-written proofs for the

same security properties as the current Signal protocol. Such KEM-based protocols can be instantiated with post-quantum KEMs from the NIST competition such as Kyber [2], which will be the first NIST PQC standard for key-establishment. However, those potential replacements for X3DH and double ratchet have so far lacked computer-verified symbolic analyses which results in a limited trust in these protocols. By contrast, some other protocols, such as WireGuard [13] and (KEM)TLS [19,8], already have computer-verified symbolic analyses for post-quantum variants, both using the Tamarin prover.

Contributions. We present the first symbolic proof of a post-quantum variant of the Signal protocol. Our model focuses on the core state machines of the two main sub-protocols of this variant: the Hashimoto-Katsumata-Kwiatkowski-Prest post-quantum X3DH handshake [12] which we refer to as *PQ-X3DH*, and the Alwen-Coretti-Dodis KEM-based double ratchet [1] that we call *KEM-Double-Ratchet*. Then we show, using the Tamarin prover, that these two protocols meet the same security properties as classical X3DH and double ratchet protocols. In addition, we prove the well-formedness of the two models, which informally means that their behavior is as expected.

Our PQ-X3DH Tamarin symbolic analysis ensures the integrity of the two exchanged messages, the authentication of users, the resistance to unknown key-share attacks and replay attacks, and other properties, such as the weak forward secrecy [15] and the key compromise attack resistance, to mitigate the leak of secret information.

With regard to KEM-Double-Ratchet, our Tamarin model ensures the integrity of all the messages, the forward secrecy, and the post-compromise recovery [1]. It is worth noting that in the particular case of Signal, post-compromise recovery is met only if the adversary is passive during the recovering process. While within the double ratchet protocol two parties can exchange a potentially infinite number of messages, we model only three exchanges, which represents the minimum number of exchanges for each security property to hold. A simple induction argument then enables us to generalize these properties to any number of exchanges. To our knowledge, our formal verification model is the first one that covers the post-compromise recovery security property.

Outline. In section 2 we present the two sub-protocols of the considered variant of the Signal protocol and their Tamarin model. Then we present in section 3 the Tamarin formalism used in our symbolic analysis, the different security properties verified, and the results of our formal verification.

2 A KEM-Based Signal Protocol

In this section, we describe the KEM-based variant of the Signal protocol that is the subject of our symbolic analysis. As explained in the introduction, the Signal protocol is separated in two sub-protocols providing different functionalities, we respect this separation for this KEM-based variant in order to facilitate its analysis and clearly identify the contribution of each sub-protocol in the security of the whole protocol. The first

sub-protocol named PQ-X3DH is used as authenticated key agreement while the second one named KEM-double-ratchet is used for secure instant messaging by refreshing the session key at each time a message is sent.

2.1 The PQ-X3DH Protocol

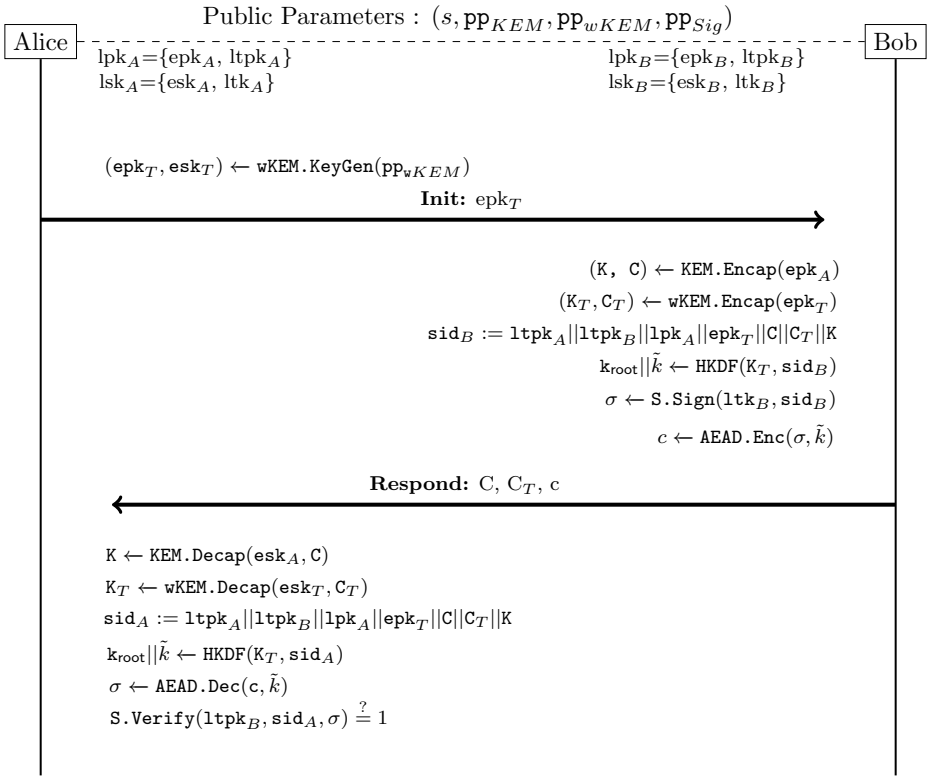

Fig. 1. The PQ-X3DH protocol.

X3DH [21] is an asynchronous protocol that generates a shared secret between the communicating parties to initialize their communication as well as authenticate themselves. It fully authenticates the receiver Bob and partially authenticates the initiator Alice. It is called asynchronous because both parties can initiate the connection while the other is offline. Such property provides flexibility but could completely break the protocol in the case of a malicious server. Apart from such a case the asynchronous protocol is highly secure. We consider here the PQ-X3DH presented in [12] which preserve the security properties of the classical protocol and we focus on the variant of PQ-X3DH that does not use a signature to fully authenticate Alice. The motivation for

this change is to allow Alice to deny having taken part in the exchange, in the same way that Bob can deny it thanks to the encryption of his signature.

The PQ-X3DH sub-protocol is presented in Figure 1. Two key encapsulation mechanisms, KEM and wKEM, are employed as building blocks in this key agreement protocol. wKEM, which is IND-CPA secure, is for ephemeral use. KEM is IND-CCA secure here. Tamarin considers the public-key encryption as ideal (thus IND-CCA), but for an ephemeral use, IND-CPA is sufficient.

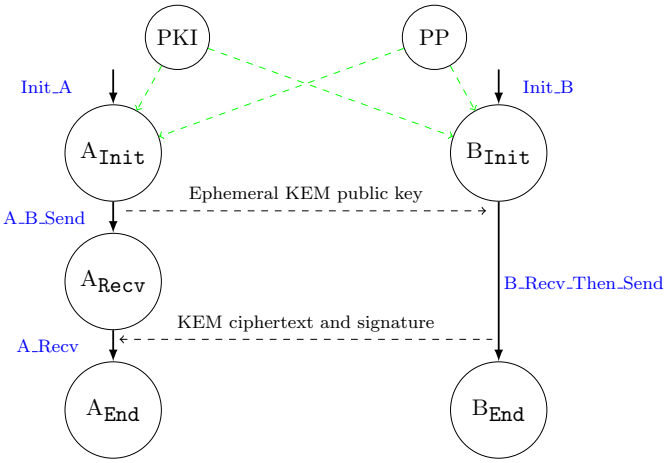

Fig. 2. Graph of the Tamarin PQ-X3DH model.

Tamarin model. Since Tamarin has no built-in KEM we replace the KEM with an asymmetric encryption scheme encrypting a fresh ephemeral key. The two approaches are equivalent considering the idealization of cryptographic primitives used in Tamarin. Our model for the PQ-X3DH protocol is represented as the state machine in Figure 2 with nine transition rules:

- `PKI` and `PP`: these rules formalize the Public Key Infrastructure (PKI), the Public Parameters (PP). PKI assigns only once a long term key with an ephemeral key to a user. Instead of handling non-replayability with a batch of ephemeral one-time keys we directly use restrictions to ensure the a key can only be used once.
- `Init_A` and `Init_B`: each user get from PKI and PP the public parameters and public keys needed for the PQ-X3DH protocol.
- `A_B_Send`: Alice sends an ephemeral public key to initiate a key encapsulation.
- `B_Recv_Then_Send`: Bob receives Alice's ephemeral public key, encapsulates two secret keys into two KEM ciphertexts, one with Alice's ephemeral public key and wKEM, the other by using Alice's long-term public key and KEM. Bob also sign the protocol transcript and send his signature encrypted with an AEAD scheme.

- `A_Recv`: Alice receives Bob's ciphertext and signature and derives a session key that will be used by Alice and Bob to communicate. Then, she decrypts and verifies the signature.
- `RevealE`: Reveals to the attacker the ephemeral secret keys.
- `RevealL`: Reveals to the attacker the long-term secret keys.

2.2 The KEM-double-ratchet Protocol

The double ratchet protocol (DR for short) is used for securing an ongoing exchange of messages between two peers by repeatedly producing fresh session keys while saving the authentication made with the PQ-X3DH initialization.

This protocol is self-healing, which means that it is made so that if at some point a user's key is intercepted by an attacker, the upcoming renewal of the session key is there to protect the secrecy of the future messages. This property is sometimes called post-compromise security. To satisfy this property, a cryptographic ratchet based on a key exchange method, such as Diffie-Hellman in the classical case, is used in the protocol, and a ratchet based on key derivation functions enables key renewal without interaction between the peers.

In order to communicate securely, the double ratchet protocol derives three types of shared secrets: *root*, *chain* and *message* secrets. They are used respectively as *master*, *derivation* and a *message* keys [20]. Since we consider a KEM-based double ratchet, we deviate a bit from this definition. As specified in Figure 3, in KEM-Double-Ratchet the two communicating peers Alice and Bob start the KEM-DR sub-protocol with a common pre-shared key $\mathsf{k}_{\mathsf{root}}$. This key comes from the key agreement protocol PQ-X3DH.

Tamarin model. As shown in Figure 4 we only perform three exchanges of the KEM-double-ratchet protocol. Three exchanges are sufficient to verify all considered security properties as discussed in section 3. Our KEM-DR model has nine transition rules:

- `Init_A` and `Init_B`: each user get a secret preshared key and Bob gets an Alice's KEM ephemeral public key.
- `Send_B1`, `Send_A`, and `Send_B2`: the user encapsulates a fresh secret key with the current KEM public key, derives a new session key, encrypts the message, then sends it encrypted with the KEM ciphertext and a new ephemeral KEM public key.
- `Recv_A1`, `Recv_B`, and `Recv_A2`: the user receives a message, derives the new session key, and verifies the integrity.
- `LeakState`: Reveals to the attacker the current user secrets.

3 Tamarin Formal Verification

In Tamarin, a protocol is seen as a state machine. A state is a multiset of facts, and rules are transitions which shift the state when some conditions are fulfilled. A rule consists of three sets of facts: *premise*, *action facts*, and *conclusion*. If all the premise

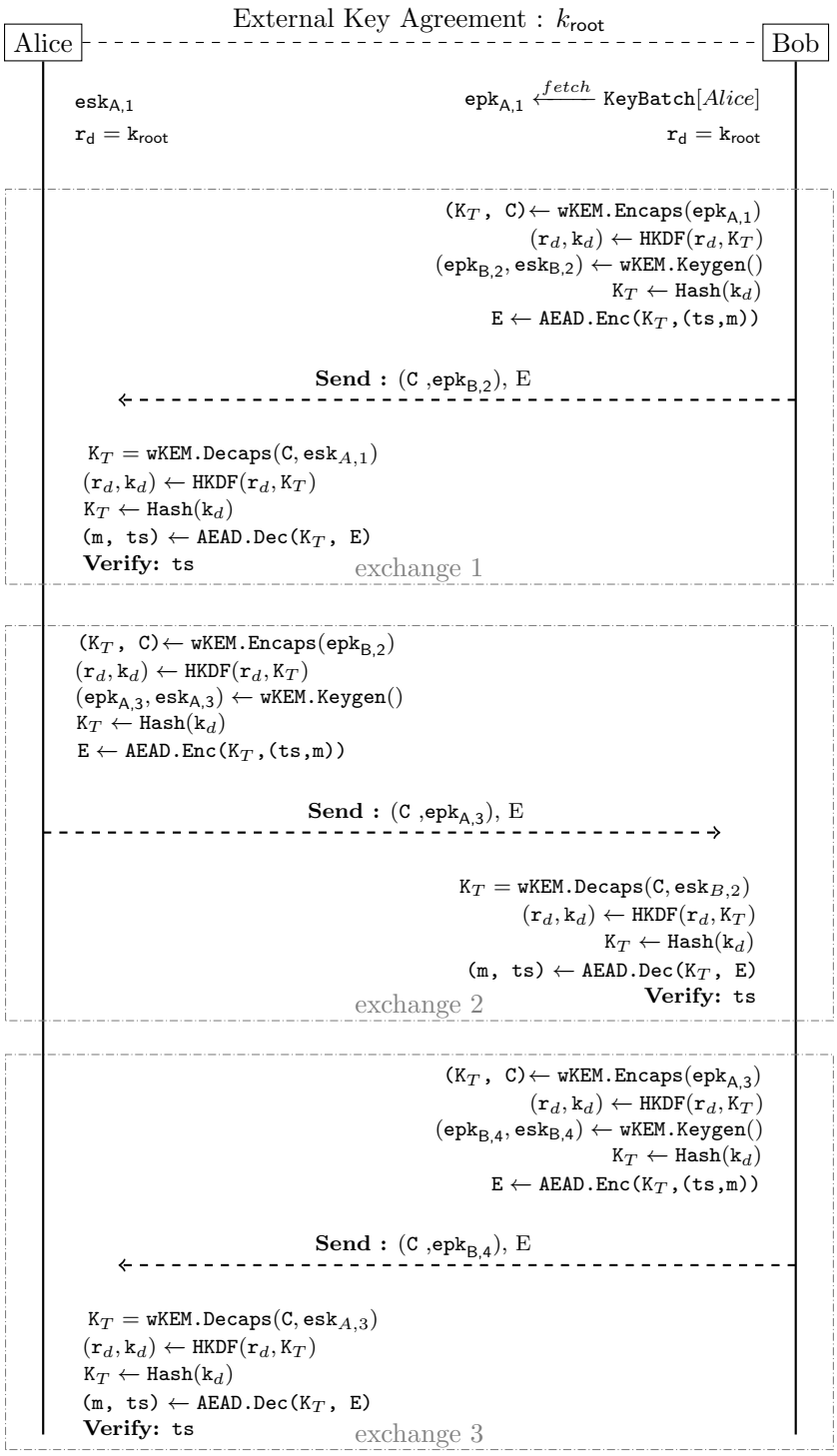

Fig. 3. The KEM-double-ratchet protocol.

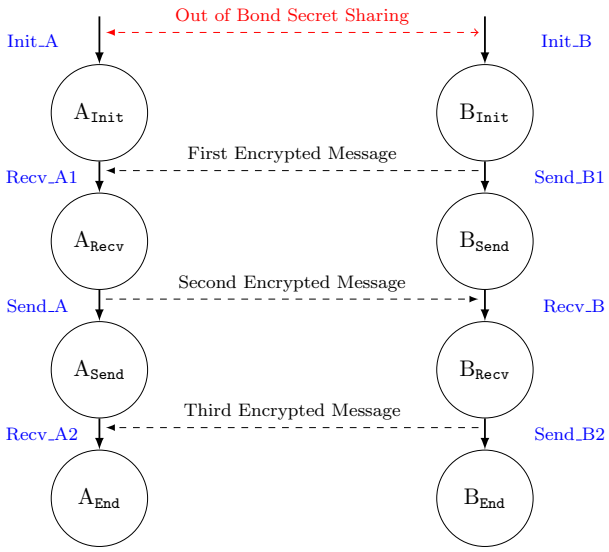

Fig. 4. Graph of the Tamarin KEM-Double-Ratchet model.

facts exist, then the rule is applied. Applying a rule means *consuming* premise facts to produce conclusion facts while recording action facts in the protocol *trace*.

Some facts are native in Tamarin such as `In()` and `Out()` to model inputs and outputs of the protocol following the Dolev-Yao model [11]. Moreover, the `Fr()` fact is used to produce fresh or unique variables.

Tamarin proposes a set of *built-ins* cryptographic primitives to model protocols, including symmetric and asymmetric encryption, hash function, and signature. It also allows to define new primitives, via `functions` and `equations` commands. In the context of this work, we define a KEM as an asymmetric encryption scheme encrypting a fresh random key, and we consider the following AEAD Tamarin formalization from [13].

In some cases, we need to restrict some transitions in the protocol, e.g., to check the equality of two terms as shown below. Hence, when a rule has the restriction `Eq(x,y)` in its action facts, then the rule is applied if and only if $x = y$.

```
restriction Eq: "All x y #i. Eq(x,y) @ #i ==> x = y"
```

Tamarin uses the logic of first order to formalize security properties as *lemmas*. The keyword `All` stands for ∀, `Ex` for ∃ and `@` represents a marker for chronological events. Lemmas use *action facts* produced by the rules to prove or disprove properties. A trivial lemma is given below, it means that if `Action1()` happened then `Action2()` happened too.

```
lemma example:
   " All x #i. Action1(x) @ #i ==> Ex y #k. Action2(y) @ #k "
```

In order to verify that a protocol has given security property, Tamarin takes as input the protocol model, with all its possible transitions as rules, and a lemma corresponding to this security property. Then, if Tamarin completes its verification process, it will either output a proof of the property or an attack trace which falsifies it.

3.1 Security Properties

The security properties verified in our symbolic analysis of PQ-X3DH and KEM-DR are the same properties as those considered in the formal verification of classical X3DH and Double-Ratchet protocols [14].

Integrity. Integrity is an important property for key exchange protocols. It allows the receiver of a message to have the assurance that the message has come unaltered from the intended sender. We separate the integrity of a message that can be verified upon receipt, called instantaneous integrity, and that which can be verified upon receipt of a subsequent message, called delayed integrity.

Authentication. We consider two authentication notions: the *partial authentication* from definition 1 and the *full authentication* from defintion 2.

Definition 1 (Partial Authentication). *A user U is partially authenticated to another user V if V can prove that the message she receives comes from the same user as the one with whom she has initialized the session.*

Definition 2 (Full Authentication). *A user U is fully authenticated to another user V if U is partially authenticated to V and V can prove the identity of U.*

Forward Secrecy. Here again we consider two different notions: the *perfect Forward Secrecy* (FS) from definition 3 and the *weak Forward Secrecy* (wFS) from definition 4. In this work, we also consider a new notion called *weak state Forward Secrecy* (wsFS) that we define as the wFS property except that the leakage concerns states instead of long term keys.

Definition 3 (perfect Forward Secrecy (FS) [18]). *A protocol is said to have* perfect forward secrecy *if compromised long-term keys does not compromise past session keys.*

Definition 4 (weak Forward Secrecy (wFS) [15]). *Any session key established by uncorrupted parties without active intervention by the attacker is guaranteed to remain secrete even if the parties to the exchange are corrupted after the session key has been erased.*

Key Compromise Impersonation Resistance. The KCI resistance from definition 5 is related to the use of long term public/private keys. Since there is no use of long term public/private key in the KEM-DR protocol, the KCI property is only applicable to the PQ-X3DH protocol.

Definition 5 (Key Compromise Impersonation (KCI) Resistance). *Even if an adversary compromises the long term private key of a user U, this adversary can not use this key to impersonate (to U) another user V that is communicating to U.*

Unknown Key-Share Resistance. We recall the definition of a UKS attack in definition 6. UKS attacks can be seen as implicit impersonation. Thus, in the same way as for the KCI resistance, this property is only applicable to the PQ-X3DH protocol.

Definition 6 (Unknown Key-Share (UKS) attack [5]). *An unknown key-share attack on an AKE protocol is an attack whereby a user U ends up believing she shares a key with V, and although this is in fact the case, V mistakenly believes the key is instead shared with an entity $W \neq U$.*

Post-Compromise Recovery. We recall the post-compromise recovery property in definition 7. By definition, this property is only applicable when a protocol is iterated repeatedly between the parties, this is the case for KEM-DR but not for PQ-X3DH.

Definition 7 (Post-Compromise Recovery (PCR) [1]). *If the attacker remains passive, i.e., the attacker does not inject any messages, and if users have access to fresh randomness, then the users recover a secure state from a compromised state after a few communication rounds.*

3.2 Tamarin Formalization

Before describing in more detail the Tamarin formalization of the different security properties presented above, we need to define some notations that will be useful later. In the rest of the paper, we use the following notations:

- $\mathcal{L} = \{0, 1, 2\}$ is the set of iteration indices, i.e., the three first exchanges ;
- $\mathcal{M} \subset \{0, 1\}^*$ is the set of messages sent via the Signal protocol ;
- $\mathcal{K} \subset \{0, 1\}^k$ is the set of secret keys where k is the key length ;
- $\mathcal{S} \subset \{0, 1\}^*$ is the set of message indices, i.e., the message numbers ;
- Σ is the set of protocol states.
- Γ is the set of user states.

Moreover, using the Tamarin formalism, we note:

- KU(x): The adversary sent x and therefore has knowledge of x ;
- K(x): The adversary has knowledge of x.

In Table 1, respectively Table 2, we introduce the Tamarin action facts and their abbreviations needed in our symbolic analysis of PQ-X3DH, respectively KEM-DR. These action facts are used to define the Tamarin lemmas corresponding to the security properties.

Table 1. Tamarin action facts for PQ-X3DH, abbreviations and definitions

Action fact	Abbreviation	Definition
SessA	S_A(A,B,k)	A accepts the key k as valid to communicate with B
ExplicitAuth	EA(B,A)	A explicitly authenticates B
RevealL	R_L(A)	The long-term key of A is revealed to the attacker
RevealE	R_E(A)	The ephemeral key of A is revealed to the attacker
SendConnect	SC(A)	A initiates the PQ-X3DH protocol
RecvConnect	RC(A,B)	B receives the initialization message from A
SendSign	SS(B,A)	B sends its signature to A
weakFS	$wFS_{A,B}$(k)	Saves k to check its resistance against future reveal
Send/Recv	$Send_{A,n}$(m)	A sends the message m of index n
	$Recv_{B,A,n}$(m)	B checks the integrity of message m of index n from A

Table 2. Tamarin action facts for KEM-Double-Ratchet, abbreviations and definitions

Action fact	Abbreviation	Definition
IntegS/IntegR	$I_{S/R}$(n,m,s)	Sends or receives message m of index n associated with session key s and checks its integrity in reception
FS	FS(A,B,n,st)	Saves the current state st associated with the sending of the message of index n between A and B in order to check its resistance against future reveal
Healed	$H_{A,B}$(st)	Checks if the state st has recovered from a previous reveal in the communication between A and B
Reveal	R(A,n)	Reveals the secrets of A associated with the message of index n.

In Tamarin, the user state is the current set of the secrets of this user. In order to characterize respectively full and partial knowledge of the user's secrets by the attacker, we define the *revealed* state in definition 8 and the *compromised* state in definition 9.

Definition 8 (Revealed State). *A state is said to be* revealed *if the adversary has knowledge of every hidden elements of the state.*

Definition 9 (Compromised State). *A state is said to be* compromised *if the adversary has knowledge of any hidden element of the state.*

PQ-X3DH Security Properties. For the sake of clarity, we describe the security properties verified with Tamarin in the usual mathematical formalism. Let E = A $<_t$ B the notation $<_t$ means that E is true if and only if event A occurs before event B. For lack of space, the definitions of the corresponding Tamarin lemmas are presented in the full version of this paper.

Integrity. The integrity is checked on both messages transmitted through the PQ-X3DH protocol. Nothing in this protocol allows an immediate integrity check of the first transmitted message. However, if both parties share the same key at the end of the protocol,

then the integrity of this message is ensured in a delayed manner. For this reason, we define the following condition under which the delayed integrity of the first message is assured:
The following properties insure, for any user A, B that have shared a common secret k by respectively sending message m_1 and receiving message m_2 on the first exchange, that $m_1 = m_2$. Which leads with Tamarin formalism too:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \forall \mathrm{m}_1, \mathrm{m}_2 \in \mathcal{M}, \forall \mathrm{k} \in \mathcal{K}.$$
$$\mathrm{S_A(A,B,k)} \wedge \mathrm{S_B(B,A,k)} \wedge \mathrm{Send_{A,1}(m_1)} \wedge \mathrm{Recv_{B,A,1}(m_2)} \implies \mathrm{m_1 = m_2} \ .$$

The integrity of the second message can be immediately verified thanks to the signature. Thus the condition to check the immediate integrity is modeled as: For any user A and B, any message m_1, m_2 if in the second flow of the exchange B sent m_1 and A received m_2 without any corruption of the long term key of B and ephemeral key of A before the reception of the second flow by A then $m_1 = m_2$. Which leads to the following Tamarin formalism:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \ \forall \mathrm{m}_1, \mathrm{m}_2 \in \mathcal{M}. \ \mathrm{Send_{B,2}(m_1)} \wedge \mathrm{Recv_{A,B,2}(m_2)} \wedge$$
$$\wedge \neg \left(\mathrm{R_E(A)} \leq_t \mathrm{Recv_{A,B,2}(m_2)}\right) \wedge \neg \left(\mathrm{R_L(B)} \leq_t \mathrm{Recv_{A,B,2}(m_2)}\right) \implies \mathrm{m_1 = m_2} \ .$$

Authentication. We consider two different notions of authentication depending on the role of the user in the PQ-X3DH protocol. Indeed, the initiator, Alice, does not sign any message and her KEM long term key is provided by a server without guaranteeing its authenticity. In these conditions, Alice can only be partially authenticated according to definition 1. In the case where the equivalent of a certificate of the Alice's KEM long term key was added, then she could be fully authenticated at the end of the PQ-X3DH protocol. The second user, Bob, signs the message which allows Alice to explicitly authenticates him under the classical conditions of a public key infrastructure. The following condition is for the full authentication of Bob by Alice:

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}. \ \mathrm{EA(B,A)} \leq_t \mathrm{R_L(B)} \implies [\mathrm{SS(B,A)} \wedge \mathrm{RC(A,B)}$$
$$\wedge \left(\mathrm{SC(A)} \wedge \mathrm{SS(B,A)} \leq_t \mathrm{EA(B,A)}\right) \wedge \left(\mathrm{SC(A)} \leq_t \mathrm{SS(B,A)}\right) \wedge \left(\mathrm{RC(A,B)} =_t \mathrm{SS(B,A)}\right)] \ .$$

This can be described as for any user A and B, if the explicit authentication has been done before any corruption on B's long term key, then the protocol has been honestly executed by A and B.

weak Forward Secrecy. The following condition verifies the wFS property on k_{root} and $\tilde{k}$ keys of the PQ-X3DH protocol in case of future compromise of the initiator's short-term and responder's signing keys.

$$\forall \mathrm{A}, \mathrm{B} \in \mathcal{U}, \forall \mathrm{k} \in \mathcal{K}. \ \neg \mathrm{R_L(B)} \leq_t \mathrm{wFS_{A,B}(k_1, k_2)} \wedge \neg \mathrm{R_E(A)} \implies \neg \mathrm{KU(k)} \ .$$

It means that for any user A and B that have agreed on a common key k, if at a certain point no corruption on the long term key of B has happened and if no corruption on the ephemeral key of A has happened then the adversary does not know the common key k.

KCI resistance. The only possible scenario in which a KCI attack occurs is when the signing long-term key of the responder is compromised, in this case it must be guaranteed that an attacker cannot use this key to impersonate any of the users. Thus we have the following condition for KCI resistance:

$$(\forall A, B, S \in \mathcal{U}, \forall k \in \mathcal{K}. S_A(A,S,k) \wedge R_L(A) \wedge S_B(B,A,k) \implies S = B) \wedge$$
$$(\forall A, B, S \in \mathcal{U}, \forall k \in \mathcal{K}. S_A(A,B,k) \wedge R_L(B) \wedge S_B(B,S,k) \implies S = A) .$$

In other word if A or B have shared a key k with a user S and if respectively A or B's long term key has been corrupted and if A or B have agreed with respectively B or A on the key k too, then respectively the user S is B or A and cannot another user.

UKS resistance. The UKS resistance consists of ensuring that if two users have agreed on a common session key, then they have the assurance that if neither key is compromised then no other user can impersonate either of them.

$$\forall A, B, C, S \in \mathcal{U}, \forall k \in \mathcal{K}. S_A(A,S,k) \wedge S_B(B,C,k) \implies S = B \wedge C = A .$$

This is a somewhat stronger approach compared to the previous properties that insure that no impersonation will occurs.

Double-Ratchet Security Properties. Regarding the KEM-DR protocol, we verify the classical security properties as well as the post-compromise recovery from [1].

Integrity. For each exchange, we verify that the message sent is indeed the message received thanks to the integrity provided by the AEAD scheme.

$$\forall n \in \mathcal{S}, \forall m_1, m_2 \in \mathcal{M}, \forall k \in \mathcal{K}. I_S(n,m_1,s) \wedge I_R(n,m_2,s) \wedge \neg K(k) \implies m_1 = m_2 .$$

PCR. As this property ensures that for a corruption during a given exchange it is enough to wait for two exchanges before the session key is secret again, it is necessary to check this condition on three consecutive exchanges. Then this base case allows to prove theorem 1 by induction.

$$\forall A, B \in \mathcal{U}, \forall st \in \Sigma.$$
$$H_{A,B}(st) \wedge R(B,0) \wedge H_{A,B}(st) <_t R(B,0) \implies \neg K(st) \vee R(B,2) \vee R(A,1) .$$

As it can be deduced this properties is proven only on the three first messages. This can be proven using Tamarin for any fixed triplet $(n, n+1, n+2)$ of ratchet exchange. However for pure theoretical insurance we prove it using induction using this case as the base case of the induction proof.

weak state Forward Secrecy. Similarly, two consecutive exchanges of the KEM-DR protocol are sufficient to prove by induction the wsFS property in theorem 2.

$$\forall A, B \in \mathcal{U}, \forall st \in \Sigma.$$
$$FS(A,B,0,st) \wedge R(A,1) \wedge FS(A,B,0,st) <_t R(A,1) \implies \neg K(st) \vee R(B,0) .$$

3.3 Formal Verification Results

In Table 3 we present the results obtained from the automatic verification with Tamarin of the security properties considered for the PQ-X3DH and KEM-DR protocols.

Table 3. Results of Tamarin verification for PQ-X3DH and KEM-Double-Ratchet protocols.

Protocols	Integrity		Auth.	Imp. resistance		Forward secrecy			PCR
	Instant	Delayed		KCI	UKS	FS	wFS	wsFS	
PQ-X3DH	✗	✓	✓	✓	✓	✗	✓	NA	NA
KEM-Double-Ratchet	✓	NA	NA	NA	NA	✓	NA	✓	✓

Since the KEM-DR protocol admits an arbitrary number of interactions, properties impacting previous or future states of the protocol require an additional proof in order to holds for any exchange of the protocol. Only the PCR and wsFS properties fall in this case, the others are trivially proven. To be more precise Tamarin allows these propreties to be true for any $k < n$ with n fixed. We therefore propose to extend it to arbitrary n by induction for theoretical purposes.

Theorem 1 (KEM-Double-Ratchet Post Compromise Security). *For all user state $State_n$ with $n > 0$:*

$$\begin{aligned} &\mathit{Compromised}(State_n) \wedge \neg\, \mathit{Revealed}(State_{n+1}) \wedge \neg\, \mathit{Revealed}(State_{n+2}) \\ &\implies \mathit{Healed}(State_{n+2})\ . \end{aligned}$$

Proof. We prove theorem 1 by induction for all integer $n > 0$. The base case has been proven using Tamarin. Suppose that the theorem is true for all integer $k < n$, and that:

$$\neg\, \texttt{Healed}(\text{State}_{n+3}) \quad \text{with} \quad \neg\, \texttt{Compromised}(\text{State}_{n+3})$$

First:

$$\texttt{Compromised}(\text{State}_{n+1}) \implies \exists k \leq n+1, \texttt{Revealed}(\text{State}_k)$$

Let name $i = \max\{k \leq n+2, \texttt{Revealed}(\text{State}_k)\}$, if $i < n+2$ then by definition of i:

$$\neg\, \texttt{Revealed}(\text{State}_{i+1}) \wedge \neg\, \texttt{Revealed}(\text{State}_{i+2})$$

and by induction hypothesis, the state $i+2$ is healed which means that the state $n+3$ is healed too since there is no reveal in between step $i+1$ and $n+3$. Now if $i = n+2$ let remind the definition of an healed state:
By definition of a healed state, for all $n > 2$ we have:

$$\begin{aligned} \texttt{Healed}(\text{State}_n) \iff \exists k < n,\ &[\texttt{Revealed}(\text{State}_k) \wedge \neg\, \texttt{Compromised}(\text{State}_n)] \\ &\wedge \neg \texttt{Revealed}(\text{State}_n) \end{aligned}$$

And thus:

$$\begin{aligned}\neg\,\texttt{Healed}(\text{State}_n) \implies & \forall k < n, [\neg\texttt{Revealed}(\text{State}_k) \vee \texttt{Compromised}(\text{State}_n)] \\ & \vee \texttt{Revealed}(\text{State}_n) \\ & \implies \forall k < n, [\neg\texttt{Revealed}(\text{State}_k) \vee \exists j \leq n, \texttt{Revealed}(\text{State}_j)] \\ & \vee \texttt{Revealed}(\text{State}_n), \text{ by definition of } \texttt{Compromised}\end{aligned}$$

Since we have $i = n + 2$ then: $\texttt{Revealed}(\text{State}_{n+2})$. Additionally:

$$\forall k \leq n, \neg\texttt{Revealed}(\text{State}_k) \iff \neg\texttt{Compromised}(\text{State}_n)$$

Therefore:

$$\begin{aligned}\neg\texttt{Healed}(\text{State}_{n+3}) \implies & \neg\texttt{Compromised}(\text{State}_{n+1}) \\ & \vee \texttt{Revealed}(\text{State}_{n+2}) \vee \texttt{Revealed}(\text{State}_{n+3})]\end{aligned}$$

Which ends our induction proof. □

We introduce the notion of *healing ball* in definition 10 to prove the wsFS property in theorem 2.

Definition 10 (Healing Ball). *We define the* healing ball B_h *for all user state* $S \in \Gamma$*, as* $B_h(S) = \{\gamma \in \Gamma \mid \texttt{Revealed}(\gamma) \implies \neg\,\texttt{Healed}(S)\}$.

Theorem 2 (KEM-Double-Ratchet weak state Forward Secrecy). *For all user state* $State_n$ *with* $n \geq 2$*:*

$$\begin{aligned}&\texttt{Compromised}(State_n) \wedge (\forall k < n,\ S \in B_h(State_k), \neg\,\texttt{Revealed}(S)) \\ &\implies \neg\,\texttt{Compromised}(State_k)\ .\end{aligned}$$

Proof. We prove theorem 2 by induction for all integer $n > 1$. The base case has been proven using Tamarin. Suppose that the theorem is true for all integer $\ell \leq n$, and that State_{n+1} has been compromised. Then, we have:

$$\texttt{Compromised}(\text{State}_{n+1}) \implies \texttt{Revealed}(\text{State}_n) \vee \texttt{Revealed}(\text{State}_{n+1})$$

and by definition, for all ℓ:

$$\texttt{Revealed}(\text{State}_\ell) \implies \texttt{Compromise}(\text{State}_\ell)$$

If State_n has been revealed and not State_{n+1}, we apply the induction hypothesis. Now suppose that State_{n+1} has been revealed but not State_n, we then use the fact that KEM.decaps is supposed ideal by Tamarin and then deterministic, so regarding the backward analysis State_{n+1} is a deterministic function of State_n. Finally, if both states have been revealed, then we apply the induction hypothesis. □

Acknowledgement

We wish to thanks Matthieu Giraud and Renaud Dubois for help on the Tamarin prover as well as helping discussions on the subject and lastly the reviewers for highlighting some typos.

References

1. Alwen, J., Coretti, S., Dodis, Y.: The double ratchet: Security notions, proofs, and modularization for the Signal protocol. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 129–158. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019).
2. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber – Submission to round 3 of the NIST post-quantum project (2021), https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf
3. Basin, D.A., Dreier, J., Hirschi, L., Radomirovic, S., Sasse, R., Stettler, V.: A formal analysis of 5g authentication. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) CCS. pp. 1383–1396 (2018)
4. Bhargavan, K., Blanchet, B., Kobeissi, N.: Verified models and reference implementations for the TLS 1.3 standard candidate. In: IEEE Symposium on Security and Privacy, SP. pp. 483–502 (2017)
5. Blake-Wilson, S., Menezes, A.: Unknown key-share attacks on the station-to-station (STS) protocol. In: Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC. vol. 1560, pp. 154–170 (1999)
6. Blanchet, B.: Modeling and verifying security protocols with the applied pi calculus and proverif. Found. Trends Priv. Secur. **1**(1-2), 1–135 (2016)
7. Brendel, J., Fischlin, M., Günther, F., Janson, C., Stebila, D.: Towards post-quantum security for signal's X3DH handshake. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 404–430. Springer (2020)
8. Celi, S., Hoyland, J., Stebila, D., Wiggers, T.: A tale of two models: Formal verification of KEMTLS via Tamarin. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds.) ESORICS 2022, Part III. LNCS, vol. 13556, pp. 63–83. Springer, Heidelberg, Germany, Copenhagen, Denmark (Sep 26–30, 2022).
9. Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the signal messaging protocol. J. Cryptol. **33**(4), 1914–1983 (2020)
10. Cremers, C., Horvat, M., Hoyland, J., Scott, S., van der Merwe, T.: A comprehensive symbolic analysis of TLS 1.3. In: CCS. pp. 1773–1788 (2017)
11. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–207 (1983)
12. Hashimoto, K., Katsumata, S., Kwiatkowski, K., Prest, T.: An efficient and generic construction for signal's handshake (X3DH): post-quantum, state leakage secure, and deniable. In: PKC. vol. 12711, pp. 410–440 (2021)

13. Hülsing, A., Ning, K.C., Schwabe, P., Weber, F., Zimmermann, P.R.: Post-quantum WireGuard. In: 2021 IEEE Symposium on Security and Privacy. pp. 304–321. IEEE Computer Society Press, San Francisco, CA, USA (May 24–27, 2021).
14. Kobeissi, N., Bhargavan, K., Blanchet, B.: Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach. In: EuroS&P. pp. 435–450 (2017)
15. Krawczyk, H.: HMQV: A high-performance secure diffie-hellman protocol. In: Shoup, V. (ed.) CRYPTO. vol. 3621, pp. 546–566 (2005)
16. Lowe, G.: Breaking and fixing the needham-schroeder public-key protocol using FDR. In: TACAS. vol. 1055, pp. 147–166 (1996)
17. Meier, S., Schmidt, B., Cremers, C., Basin, D.A.: The TAMARIN prover for the symbolic analysis of security protocols. In: Computer Aided Verification CAV. vol. 8044, pp. 696–701 (2013)
18. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
19. Schwabe, P., Stebila, D., Wiggers, T.: Post-quantum TLS without handshake signatures. In: CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event. pp. 1461–1480 (2020)
20. Trevor Perrin, M.M.: The double ratchet algorithm, https://signal.org/docs/specifications/doubleratchet/
21. Trevor Perrin, M.M.: The x3dh key agreement protocol, https://signal.org/docs/specifications/x3dh/