The Hierarchy of Key Evolving Signatures and a Characterization of Proxy Signatures

Tal Malkin[1], Satoshi Obana[2] and Moti Yung[1]

[1] Columbia University
{tal,moti}@cs.columbia.edu
[2] NEC and Columbia University
obana@bx.jp.nec.com

Abstract. For the last two decades the notion and implementations of *proxy signatures* have been used to allow transfer of digital signing power within some context (in order to enable flexibility of signers within organizations and among entities). On the other hand, various notions of the key-evolving signature paradigms (forward-secure, key-insulated, and intrusion-resilient signatures) have been suggested in the last few years for protecting the security of signature schemes, localizing the damage of secret key exposure.

In this work we relate the various notions via direct and concrete security reductions that are tight. We start by developing the first formal model for fully hierarchical proxy signatures, which, as we point out, also addresses vulnerabilities of previous schemes when self-delegation is used. Next, we prove that proxy signatures are, in fact, equivalent to key-insulated signatures. We then use this fact and other results to establish a tight hierarchy among the key-evolving notions, showing that intrusion-resilient signatures and key-insulated signatures are equivalent, and imply forward-secure signatures. We also introduce other relations among extended notions.

Besides the importance of understanding the relationships among the various notions that were originally designed with different goals or with different system configuration in mind, our findings imply new designs of schemes. For example, many proxy signatures have been presented without formal model and proofs, whereas using our results we can employ the work on key-insulated schemes to suggest new provably secure designs of proxy signatures schemes.

1 Introduction

Characterizing relationships among cryptographic notions is an important task that increases our understanding of the notions and can contribute to concrete designs. In this work we look at two paradigms, proxy signatures and key-evolving signatures, that were suggested at different times for totally different purposes. After developing the first formal model for fully hierarchical proxy signatures and addressing a vulnerability in previous proxy schemes, we prove that proxy signatures are equivalent in a very strong sense to key-insulated signatures (one of the key-evolving notions). We also relate the various notions within the key-evolving paradigm, that were originally suggested for different system architecture settings and adversarial assumptions, establishing a tight hierarchy among them (tight in the sense of no security loss in the reductions). In the rest of the introduction we elaborate on these primitives, our results, and their significance.

Proxy Signatures and Our Contributions in Modeling them. The paradigm of proxy signature is a method for an entity to delegate signing capabilities to other participants so that they can sign on behalf of the entity within a given context (the context and limitations on proxy signing capabilities are captured by a certain warrant issued by the delegator which is associated with the delegation act). For example, Alice the executive might want to empower Bob the secretary to sign on her behalf for a given week when Alice is out of town. Such proxy capability transfer may be defined recursively to allow high flexibility in assigning limited entitlements. The notion is motivated by real life flexibility of "power of attorney" and other mechanisms of proxy.

The notion has been suggested and implemented in numerous works for about 20 years now: one of the early works to be published was presented in [6], whereas for a cryptographic treatment see [14]. Most of the past work is informal and without complete proofs. The first (and to the best of our knowledge, only) work to formally define the model of proxy signatures, is the recent work of Boldyreva, Palacio, and Warinschi [3]. Their definition is of proxy signature, with only one level of delegation, and without using the warrants as part of the model (though warrants are used in the common scheme of delegation by certificate, a notion that was analyzed by [3]).

We provide the first definition of fully hierarchical proxy signatures with warrants, supporting chains of several levels of delegation. Furthermore, the fully hierarchical approach illuminates an important aspect of proxy signatures, regarding self-delegations, which was previously overlooked. Specifically, we identify a vulnerability in previous solutions (both in existing proxy signature implementations such as the delegation by certificate, and in the formal model which rendered them secure). This weakness, which results in enabling a delegatee to possibly take "rogue actions" on behalf of a delegator, does not exist in our model, and we point out how the delegation by certification implementation (and other schemes with the same problem) can be modified in a simple way so as to avoid such attacks, and satisfy our strong notion of security.

Key Evolving Signatures. The paradigm of key evolving signatures started with Anderson's suggestion in [1], towards mitigating the damage caused by key exposure, one of the biggest threats to security of actual cryptographic schemes. Indeed, if the secret key in a standard signature scheme is exposed, this allows for forgery, invalidation of past and future signatures, and thus repudiation through leaking of the secret key. To limit the damage, the key evolving paradigm splits the time for which the signature is valid (say, 5 years) into well defined short periods (say months, days, or a period per signature, as required by the application). The secret key can then evolve with the periods (see details below), while maintaining the same public key. This idea gave rise to three well-defined notions of protection against key exposure, compartmentalizing the damage. The three notions have different configurations and different adversarial settings, achieving different properties:

1. Forward-Secure Signature Schemes (FS) [1, 2]: Here the system is comprised of a single agent holding the private signing key, and at each period the key is evolved (via a one-way transformation) so that the exposure does not affect past periods. This notion has the advantage that even if *all* the key material is completely exposed, past signatures are still valid, and cannot be forged or repudiated. On the other hand, such a complete exposure necessarily compromises the validity of all future signatures, and the public key cannot be used any more.
2. Key-Insulated Signature Scheme (KI) [5]: Here the system is made out of two entities: the signer and a helper (base). At the start of the period the signer is updated by the helper to produce the next period's key. The helper is involved only in the updates. In fact, the helper can give the signer access to any period at any time (random access capability). The exposure of up to t of the N periods, chosen adaptively by the adversary, still keeps any period that was not exposed secure. The limitation of necessarily exposing all future keys, as in forward security does not apply anymore; this limitation is removed by the introduction of the helper (base) which is never exposed. The optimal t achieved by some of the schemes is $N-1$ where the remaining period is still secure. Note that here the keys at the helper and the signer are not forward-secure. This model was first considered in [4]. We remark that the notion of strong KI which protects the signer from the helper is irrelevant here (and there is a simple transformation from KI to strong KI).
3. Intrusion-Resilient Signature Scheme (IR) [9]: Here the scheme is also made out of a signer and a helper (base). Now the exposures of both the helper and the signer are allowed. If the exposure is alternating (i.e., at each period at most one of the signer or the helper is exposed) then the scheme remains secure for all unexposed signing periods. If the exposure is of both the helper and the signer simultaneously, then the system becomes forward-secure from that period on: the past is protected (excluding the periods where the signer was directly exposed) but the future is now necessarily insecure. Note that unlike KI, this notion allows exposure of the helper, and that both the helper's key and the signer's key are forward-secure.

Our Reductions: A Characterization of Proxy Signatures, and The Hierarchy of Key Evolving Signatures. Our goal is to explore the relations among the key evolving signature notions and proxy signatures, towards gaining a better understanding of the primitives, and obtaining practical constructions. From a complexity-theoretic point of view, one can establish equivalences using the fact that these notions have implementations based on a generic signature scheme (typically less efficient than implementations based on specific number theoretic assumptions). For example, see the generic constructions of [2, 12, 5, 7] for key evolving signatures, and the delegation by certificate scheme for proxy signatures that was suggested with different variations in numerous works (see Section 2.1). Thus, the notions are equivalent to the existence of one-way functions in terms of computational assumptions [15, 16]. However, our goal is to establish *direct* reductions, both from a practical point of view (namely, given an implementation of one primitive, construct the other primitive using the first almost "as-is", with a straight-forward and efficient transformation), and from a theoretical point of view: analyzing the efficiency and the *concrete* security guarantees. In particular, we consider direct reductions between paradigms so that there is a concrete security evaluation of one scheme based on the concrete security of the related scheme to which it is reduced, while minimizing the loss of the concrete security value, and minimizing overhead in efficiency. Under this notion of direct reduction we found that:

- Proxy signatures are equivalent to KI signatures. In particular, we show that proxy signatures imply KI signatures via a tight reduction achieving the same concrete security, and that KI signatures imply proxy signatures via a tight security reduction. Our characterization of proxy signatures immediately provides a suite of provably secure proxy signature schemes, based on the previous (and future) schemes for KI signatures. For example, all the schemes of [5] can be used, including the efficient ones based on trapdoor-signature schemes, and their instantiations (based on RSA, identity-based signatures from the Gap Diffie-Hellman group, etc.). This is a significant contribution, since only few provably secure proxy schemes were known before (e.g., [3] for the non-hierarchical case).
- We show a direct and tight hierarchy for key evolving signature schemes. Specifically, we show that IR implies KI implies FS, and KI implies IR without loss in concrete security. The implication KI $\rightarrow$ FS was left as an open problem in [5], and our proof of it utilizes our result about the equivalence of KI and proxy signatures.[1] Note that while proving IR$\rightarrow$ FS is trivial, relating them to KI is not. For example, the naive approach of unifying the signer and helper of the KI model into the single signer entity of the FS model, does not work. This is because the keys of the signer and helper together are *not* forward-secure, by definition. In fact, the opposite is true since the helper keys with the signing key for any period should be able to provide the signing key for all other periods through the random-access property.

The relationships we establish are summarized in Figure 1 on the left side. In addition, on the right side is a diagram summarizing our technical results which are employed in the derivation of these relationships, showing the structure of our proofs (and may be helpful to obtain the best constructions from an engineering

[1] Once we established this result through the connection to proxy signatures, we also succeeded in showing that KI $\rightarrow$ IR, which together with the trivial IR $\rightarrow$ FS gave an alternative proof that KI $\rightarrow$ FS directly within key evolving signatures.

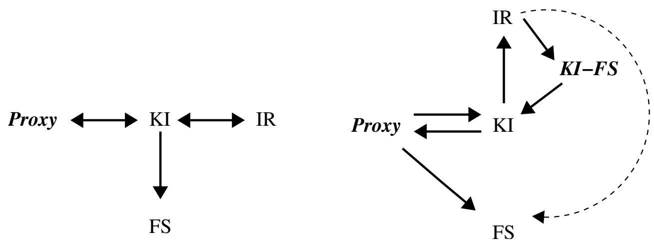

Fig. 1. The left diagram is a summary of our main results, and the right diagram is a summary of our technical reductions used to established them.

point of view). In particular, we introduce an intermediate notion between IR and KI, denoted KI-FS, which has helped us to conceptualize the IR $\rightarrow$ KI relation (and it may be of independent interest for certain applications). The dashed line refers to the trivial implication of FS from IR, which together with our result that KI implies IR gives an alternative proof that KI implies FS. We believe that directly relating proxy signing (which is a trust management and flexibility mechanism) to that of key evolving signatures (which are mechanisms to protect against key exposure attacks) is somewhat surprising and conceptually interesting. This was also a crucial step in answering the question about the relation between KI and FS, a recognized open question about seemingly closer notions.

Organization We provide the definitions for proxy signature schemes in Section 2.1, together with motivations and discussions of the model. This includes the differences and generalizations of our model compared with the previous single-level model, the weakness of previous schemes, how it is addressed by our model, and how to modify previous schemes to achieve security. In Section 2.2 we briefly review definitions for the key-evolving notions of IR, KI, and FS. In Section 3 we present the characterization of proxy signatures as equivalent to KI. Finally, in Section 4 we present the hierarchy of key evolving signatures, by showing that IR implies KI (which is a consolidation of our proofs that IR implies KI-FS and that KI-FS implies KI, given in the full version), KI implies IR, and by showing that Proxy implies FS (and therefore KI implies FS).

2 Definitions of Proxy Signatures and Key Evolving Signatures

2.1 Proxy Signature

Model Proxy signature scheme $\Pi_{\mathsf{PS}} = (\mathsf{Gen}_{\mathsf{PS}}, \mathsf{Sign}_{\mathsf{PS}}, \mathsf{Vrfy}_{\mathsf{PS}}, (\mathsf{Dlg}_{\mathsf{DPS}}, \mathsf{Dlg}_{\mathsf{PPS}}), \mathsf{PSig}_{\mathsf{PS}}, \mathsf{PVrf}_{\mathsf{PS}}, \mathsf{ID}_{\mathsf{PS}})$ consists of the following eight algorithms.

$\mathsf{Gen}_{\mathsf{PS}}$, the key generation algorithm, which takes security parameters $k \in \mathbb{N}$ as input, output an signing key SK and a public key PK.

$\mathsf{Sign}_{\mathsf{PS}}$, the signing algorithm, which takes a signing key SK and a message M as input, outputs a signature sig on M.

$\mathsf{Vrfy}_{\mathsf{PS}}$, the verification algorithm, which takes the public key PK, a message M, and a candidate signature sig as input, outputs a bit b, where $b = 1$ iff the signature is accepted.

$(\mathsf{Dlg}_{\mathsf{DPS}}, \mathsf{Dlg}_{\mathsf{PPS}})$, (interactive) proxy-designation algorithms (where $\mathsf{Dlg}_{\mathsf{DPS}}$ and $\mathsf{Dlg}_{\mathsf{PPS}}$ are owned by the designator i_{L-1} and the proxy signer i_L, respectively.)

$\mathsf{Dlg}_{\mathsf{DPS}}$ takes public keys of a designator $PK_{i_{L-1}}$ and a proxy signer PK_{i_L}, the signing key of which the designator delegates its signing right (i.e., the signing key is either a signing key $SK_{i_{L-1}}$ or a proxy signing key $SKP_{i_0 \dashrightarrow i_{L-1}}$ depending on whether i_{L-1} is original signer or proxy signer), a warrant up to previous delegation W_{L-1} and a warrant ω_L set in current delegation as inputs. $\mathsf{Dlg}_{\mathsf{DPS}}$ has no local output. Note that the warrant usually contains the information on "valid period", "limitation", etc. We say that a message *violates* a warrant if the message is not compliant with the contents of the warrant.

$\mathsf{Dlg}_{\mathsf{PPS}}$ takes public keys of a designator $PK_{i_{L-1}}$ and a proxy signer PK_{i_L}, the secret key of the proxy signer SK_{i_L} as inputs and outputs a proxy signing key $SKP_{i_0 \dashrightarrow i_L}$ and a warrant W_L. Note that no secret key is given when the type of the designation is "self delegation" in which the designator designates its signing right to itself with limited capability[2].

$\mathsf{PSig}_{\mathsf{PS}}$, the proxy signing algorithm, which takes a proxy signing key $SKP_{i_0 \dashrightarrow i_L}$, a message M and a warrant W as input, outputs a proxy signature $psig$.

$\mathsf{PVrf}_{\mathsf{PS}}$, the proxy verification algorithm, which takes a public key PK_{i_0} of the original designator, a message M, a warrant W, and a proxy signature $psig$ as input, outputs a bit b, where $b = 1$ iff the proxy signature is accepted.

$\mathsf{ID}_{\mathsf{PS}}$, the proxy identification algorithm, which takes a warrant W and a proxy signature $psig$ as input, outputs a list of identity (i.e., public key) PK^* in the delegation chain.

CORRECTNESS: We require that all message M and any delegation chain $j_0 \rightarrow j_2 \rightarrow \cdots \rightarrow j_L$, $\mathsf{PVrf}_{\mathsf{PS}}(PK_{i_0}, M, W_L, \mathsf{PSig}_{\mathsf{PS}}(SKP_{i_0 \dashrightarrow i_L}, M, W_L)) = 1$ and $\mathsf{ID}_{\mathsf{PS}}(W, \mathsf{PSig}_{\mathsf{PS}}(SKP_{i_0 \dashrightarrow i_L}, M, W_L)) = (PK_{i_0}, \ldots, PK_{i_L})$ if the proxy signing key $SKP_{i_0 \dashrightarrow i_L}$ and the warrant W_L is the output of consecutive executions of $(SKP_{i_0 \dashrightarrow i_l}, W_l) \leftarrow \begin{bmatrix} \mathsf{Dlg}_{\mathsf{DPS}}(PK_{i_{l-1}}, PK_{i_l}, SK_{i_{l-1}}, W_{l-1}, \omega_l), \\ \mathsf{Dlg}_{\mathsf{DPS}}(PK_{i_{l-1}}, PK_{i_l}, SK_{i_l}) \end{bmatrix}$; and the message M does not violate the warrant W_L.

Definition of Security Let F be a probabilistic polynomial-time oracle Turing machine with the following oracles:

– O_{sig}, the signing oracle, which

[2] This is significant since if the proxy signer (or device) has the original signing key in self delegation it is impossible for the designator to limit the signing capability of the proxy signer.

1. on input ("s", M, j), outputs $\mathsf{Sign}_{\mathsf{PS}}(SK_j, M)$.
2. on input ("p", $M, (j_1, \ldots, j_L), W$), outputs $\mathsf{PSig}_{\mathsf{PS}}(SKP_{j_1 \dashrightarrow j_L}, M, W)$.

- O_{sec}, the key exposure oracle, which on input
  1. ("s", j), outputs (SK_j, PK_j).
  2. ("sd", $j, L, (\omega_1, \ldots, \omega_{L-1})$), outputs the pair of self proxy signing key and the warrant $(SKP_{j \dashrightarrow j}, W)$ where the length of the delegation chain is L.
- O_{Dlg}, the designation oracle, which
  1. on input ("d", $(j_1, \ldots, j_L, W, \omega)$), interacts with $\mathsf{Dlg}_{\mathsf{PPS}}(PK_{j_{L-1}}, PK_{j_L}, SK_{j_L})$ on behalf of $\mathsf{Dlg}_{\mathsf{DPS}}(PK_{j_{L-1}}, PK_{j_L}, SKP_{j_1 \dashrightarrow j_{L-1}}, W, \omega)$.
  2. on input ("p", $(j_1, \ldots, j_L)$), interacts with $\mathsf{Dlg}_{\mathsf{DPS}}(PK_{j_{L-1}}, PK_{j_L}, SKP_{j_1 \dashrightarrow j_{L-1}}, W, \omega)$ on behalf of $\mathsf{Dlg}_{\mathsf{PPS}}(PK_{j_{L-1}}, PK_{j_L}, SK_{j_L})$.

Let $Q = (Q_{\mathsf{sec}}, Q_{\mathsf{Dlg}})$ where Q_{sec} and Q_{Dlg} be the set of F's valid query to the key exposure oracle and designation oracle, respectively. We say that the scheme is

- (j, Q)-signable if and only if ("s", j) $\in Q_{\mathsf{sec}}$.
- $((j_1, \ldots, j_L), W, Q)$-proxy-signable if and only if either of the following holds
  1. ("s", j) $\in Q_{\mathsf{sec}}$ (for all j such that $1 \le j \le L$)
  2. there exists $L'(\le L)$ such that
     - ("d", $(j_1, \ldots, j_{L'}, W', \omega')$) $\in Q_{\mathsf{Dlg}}$
     - $W', (W', \omega')$ do not contradict W
     - $j_l = j_{l-1}$ or ("s", j_l) $\in Q_{\mathsf{sec}}$ (for $L' \le l \le L$)
  3. there exists $L'(\le L)$ such that
     - $j_1 = \ldots, j_{L'}$ and ("sd", $L', (\omega_1, \ldots, \omega_{L'-1})$) $\in Q_{\mathsf{sec}}$
     - ω_i do not contradict W
     - $j_l = j_{l-1}$ or ("s", j_l) $\in Q_{\mathsf{sec}}$ (for $L' < l \le L$)

Let $\mathsf{Succ}_F^{\Pi_{\mathsf{PS}}}(k)$ be defined as follows,

$$\mathsf{Succ}_F^{\Pi_{\mathsf{FS}}}(k) = \Pr\left[\begin{array}{l} (\sigma = (M, s, PK) \wedge \mathsf{Vrfy}_{\mathsf{PS}}(PK, M, s) = 1) \quad \vee \\ (\sigma = (M, W, ps, PK) \wedge \mathsf{PVrf}_{\mathsf{PS}}(PK, M, ps) = 1) \end{array} \middle| \begin{array}{c} (SK_j, PK_j) \leftarrow \mathsf{Gen}_{\mathsf{PS}}(1^k) \\ \sigma \leftarrow F^{O_{\mathsf{sig}}, O_{\mathsf{sec}}, O_{\mathsf{Dlg}}}(1^k) \end{array}\right]$$

where

- M is never queried to O_{sig} and the scheme is not (j, Q)-signable if $\sigma = (M, s, PK_j)$.
- $((i_1, \ldots, i_L), M, W)$ is never queried to O_{sig} and the scheme is not $((i_1, \ldots, i_L), W, Q)$-proxy-signable if ("s", i_L) $\notin Q_{\mathsf{sec}}$ where $\sigma = (M, W, ps, PK)$ and $\mathsf{ID}_{\mathsf{PS}}(W, ps) = (PK_{i_1}, \ldots, PK_{i_L})$.

We say Π_{PS} is (τ, ϵ, q)-secure proxy signature if $\mathsf{Succ}_F^{\Pi_{\mathsf{PS}}}(k) < \epsilon$ for any probabilistic polynomial time Turing machine F with running time at most τ and the number of the queries to O_{sig} is upper bounded by q.

Discussion: Delegation by Certificate, and the Self-Delegation Attack Delegation by certificate is a well-known simple notion. It achieves delegation by the fact that the delegator computes a warrant $W = \mathsf{Sign}(SK_d, (PK_p, \mathtt{limitation}))$ with its secret key where SK_d is the secret key of the delegator and PK_p is the public key of the proxy signer. The proxy signer can computes a proxy signature ps for the message M simply by $ps = \mathsf{Sign}(SK_p, (W, M))$.

Delegation by certificate works well in many setting, however, we must be aware that a naive implementation leads to an attack, even on the delegation by certificate scheme. Specifically, we must take care of implementing *self-delegation* securely. For example, the scheme in [3] is not secure under our security definition, and it can be easily broken simply by querying ("sd", $2, \Lambda$) from the Q_{sig} oracle (we will use Λ to denote null data.) Since the scheme of [3] is constructed in such a way that the proxy signing key is exactly the same as the original signing key of the proxy signer even in the case of self-delegation, an adversary can forge (non-proxy) signature for any message simply by querying the self-delegation signing key. We must carefully consider the meaning of the self-delegation, which is usually used for delegating *limited* signing capability.

The model proposed in [3] also possesses the problem of self-delegation. Namely, the oracles defined by [3] only allows giving transcript of $\mathsf{Dlg}_{\mathsf{DPS}}$ and $\mathsf{Dlg}_{\mathsf{PPS}}$. Therefore, there is no way for the adversary to get the self-delegation key. This is not the case in real life since self-delegation is needed when the signing key is stored in insecure environment (e.g. laptop PC get delegation from a host). Therefore, the scheme must be secure even if the self-delegation key is exposed. In contrast, our model allows the adversary to gain self-delegation keys to reflect this real life setting. Our implementation of proxy signature based on KI also takes care of this problem. Namely, in our implementation, new key pair is always generated in self-delegation, which prevents the attack above.

In defining the model of proxy signatures the most crucial point is how to treat the semantics of the warrant since the warrant usually contains application specific information. Therefore, in the model level, it is desirable not to define the detailed semantics. In our model no semantics is defined for the warrant, it is only defined as input and output of the algorithm and a messages can be in agreement or in violation with the warrant. Further, not having access to a warrant prevents the usage of the delegated key, which is part of our model.

We also note that, in the general case, the chain of warrants may have arbitrary information in it and one needs to read it to understand whether a message is in agreement with the warrant. In this cases the length of verification of a proxy signature must be linear in the size of the delegation chain. (Of course, if warrants are of special semantics, e.g if they are not present at all, then this may be improved, e.g using aggregate signatures as suggested by [3].)

2.2 Definitions of Key-Evolving Signatures

In this section we briefly review the definition of key-evolving signatures. These definitions are the same ones as introduced in the original papers, except that we unify them following the notations of [9]. The complete definition of each notion is given in the full version of this paper [13].

Model The three key-evolving signature notions, Forward-Secure signatures (FS), Key-Insulated signatures (KI), and Intrusion-Resilient signatures (IR) consist of subsets of the following seven algorithms, as indicated below.

Gen, the key generation algorithm, which takes security parameters $k \in \mathbb{N}$ and the total number of periods N as input, outputs an initial signing key SK_0 and public key PK. Gen also outputs initial key for the helper (base) SK^* in KI and IR.

Upd^*, the update algorithm of the base, which takes a base key SK^* as inputs, key update message SKU. Key update message is used to update the signing key of the signer SK. In KI, indices i, j is also taken as input where i denotes the current time period of SK and j denotes the time period of SK after update. In IR, base key is also updated by Upd^*.

Upd, the signer key update algorithm, which takes a signer key SK_i of the previous time period i and a key update message SKU, outputs the signer key SK_j of the time period j. In IR, j is always $i+1$ whereas, in KI, i and j can be chosen arbitrary within the condition $0 \leq i, j \leq N$.

Refr^*, the base-key refresh algorithm, which takes a base key SK^* of the current time period, outputs new base key of the current time period and a key refresh message SKR. Only IR has Refr^*.

Refr, the signer-key refresh algorithm, which takes a signer key SK of the current time period and a key refresh message SKR, outputs new signer key of the current time period. Only IR has Refr.

Sign, the signing algorithm, which takes a signer key SK_j, an index of a time period j and a message M as input, outputs a signature $\langle j, sig \rangle$ on M for time period j.

Vrfy, the verification algorithm, which takes the public key PK, a message M, a pair $\langle j, s \rangle$, outputs a bit b, where $b = 1$ iff the signature is accepted.

FS consists of four algorithms $\Pi_{\mathsf{FS}} = (\mathsf{Gen}_{\mathsf{FS}}, \mathsf{Upd}_{\mathsf{FS}}, \mathsf{Sign}_{\mathsf{FS}}, \mathsf{Vrfy}_{\mathsf{FS}})$, KI consists of five algorithms $\Pi_{\mathsf{KI}} = (\mathsf{Gen}_{\mathsf{KI}}, \mathsf{Upd}^*_{\mathsf{KI}}, \mathsf{Upd}_{\mathsf{KI}}, \mathsf{Sign}_{\mathsf{KI}}, \mathsf{Vrfy}_{\mathsf{KI}})$ and IR consists of seven algorithms $\Pi_{\mathsf{IR}} = (\mathsf{Gen}_{\mathsf{IR}}, \mathsf{Upd}^*_{\mathsf{IR}}, \mathsf{Upd}_{\mathsf{IR}}, \mathsf{Refr}^*_{\mathsf{IR}}, \mathsf{Refr}_{\mathsf{IR}}, \mathsf{Sign}_{\mathsf{IR}}, \mathsf{Vrfy}_{\mathsf{IR}})$, with appropriate (and natural) correctness requirements.

Definition of Security To define the definition of security, we consider a probabilistic polynomial-time oracle Turing machine F with the following oracles:

- O_{sig}, the signing oracle, which take the message M the time period i outputs the signature for M of the designated time period.
- O_{sec}, the key exposure oracle, which on input the name of the target key (e.g. signing key, base key, etc.) and the time period, outputs the key of the designated period.

Let Q be the set of valid key exposure query of F. Then the successful probability of the adversary $\mathsf{Succ}_F^{\Pi_{\mathsf{FS}}}(k)$ can be defined as follows,

$$\mathsf{Succ}_F^{\Pi}(k) = \Pr\left[\mathsf{Vrfy}_{\mathsf{FS}}(PK, M, \langle i, s\rangle) = 1 \,\middle|\, \begin{array}{c}(PK, SK_0) \leftarrow \mathsf{Gen}(1^k),\\ (M, \langle i, s\rangle) \leftarrow F^{O_{\mathsf{sec}}, O_{\mathsf{sig}}}(PK)\end{array}\right]$$

where (M, i) is never queried to O_{sig} and

- (In FS), F never gets the signing key before the time period i.
- (In KI), F never gets the signing key of the time period i.
- (In IR), F never gets the signer key of the time period i and F never gets the signer key and the base key simultaneously in time time period before i.

We refer the reader to our full version [13] (and to the original papers [2, 5, 9]) for more complete definitions.

3 Characterization of Proxy Signatures

In this section we give the characterization of proxy signature. Namely, we prove that proxy signatures are equivalent to key-insulated signatures by constructing a key-insulated signature based on any proxy signature with concrete security reduction and vice versa.

3.1 Proxy $\rightarrow (N-1, N)$ KI

We construct $(N-1, N)$ key-insulated signature as follows. The signing key of time period j corresponds to proxy signing key with delegation chain of length $j+1$. The important point is that the proxy signer is changed every time when the period changes, which prevents the attacker who gets the signing key of period j from forging the signature of the other periods.

The complete construction of $\Pi_{\mathsf{KI}} = (\mathsf{Gen}_{\mathsf{KI}}, \mathsf{Upd}^*_{\mathsf{KI}}, \mathsf{Upd}_{\mathsf{KI}}, \mathsf{Sign}_{\mathsf{KI}}, \mathsf{Vrfy}_{\mathsf{KI}})$ from proxy signature $\Pi_{\mathsf{PS}} = (\mathsf{Gen}_{\mathsf{PS}}, \mathsf{Sign}_{\mathsf{PS}}, \mathsf{Vrfy}_{\mathsf{PS}}, (\mathsf{Dlg}_{\mathsf{DPS}}, \mathsf{Dlg}_{\mathsf{PPS}}), \mathsf{PSig}_{\mathsf{PS}}, \mathsf{PVrf}_{\mathsf{PS}}, \mathsf{ID}_{\mathsf{PS}})$ is as follows.

$\underline{\underline{\mathsf{Gen}_{\mathsf{KI}}(1^k, N)}}$

$(SK_*^{(\mathsf{PS})}, PK_*^{(\mathsf{PS})}) \leftarrow \mathsf{Gen}_{\mathsf{PS}}(1^k)$; $\quad (SK_0^{(\mathsf{PS})}, PK_0^{(\mathsf{PS})}) \leftarrow \mathsf{Gen}_{\mathsf{PS}}(1^k)$;

$(SKP_{*\to 0}^{(\mathsf{PS})}, W_0) \leftarrow \begin{bmatrix}\mathsf{Dlg}_{\mathsf{DPS}}(PK_*^{(\mathsf{PS})}, PK_0^{(\mathsf{PS})}, SK_*^{(\mathsf{PS})}, \Lambda, \Lambda),\\ \mathsf{Dlg}_{\mathsf{PPS}}(PK_*^{(\mathsf{PS})}, PK_0^{(\mathsf{PS})}, \Lambda)\end{bmatrix}$;

$SK^{*(\mathsf{KI})} \leftarrow (PK_*^{(\mathsf{PS})}, SK_*^{(\mathsf{PS})})$; $\quad SK_0^{(\mathsf{KI})} \leftarrow (SKP_{*\to 0}^{(\mathsf{PS})}, W_0)$;

$PK^{(\mathsf{KI})} \leftarrow PK_*^{(\mathsf{PS})}$;

output $(SK^{*(\mathsf{KI})}, SK_0^{(\mathsf{KI})}, PK^{(\mathsf{KI})})$;

$\underline{\underline{\mathsf{Upd}^*_{\mathsf{KI}}(SK^{*(\mathsf{KI})}, i, j)}}$

$(PK_*^{(\mathsf{PS})}, SKP) \leftarrow SK^{*(\mathsf{KI})}$;
$W \leftarrow \Lambda$;
`for` $n = 0$ `to` j `do`

$$(SKP, W) \leftarrow \begin{bmatrix} \mathsf{Dlg}_{\mathsf{DPS}}(PK_*^{(\mathsf{PS})}, PK_*^{(\mathsf{PS})}, SKP, W, \Lambda), \\ \mathsf{Dlg}_{\mathsf{PPS}}(PK_*^{(\mathsf{PS})}, PK_*^{(\mathsf{PS})}, \Lambda) \end{bmatrix};$$

$(SK_j^{(\mathsf{PS})}, PK_j^{(\mathsf{PS})}) \leftarrow \mathsf{Gen}_{\mathsf{PS}}(1^k)$;

$$(SKP, W) \leftarrow \begin{bmatrix} \mathsf{Dlg}_{\mathsf{DPS}}(PK_*^{(\mathsf{PS})}, PK_j^{(\mathsf{PS})}, SKP, W, \Lambda), \\ \mathsf{Dlg}_{\mathsf{PPS}}(PK_*^{(\mathsf{PS})}, PK_j^{(\mathsf{PS})}, SK_j^{(\mathsf{PS})}) \end{bmatrix};$$

$SK'^{(\mathsf{KI})}_{i,j} \leftarrow (SKP, W)$;

output $SK'^{(\mathsf{KI})}_{i,j}$;

$\underline{\underline{\mathsf{Upd}_{\mathsf{KI}}(SK_i^{(\mathsf{KI})}, SK'^{(\mathsf{KI})}_{i,j})}}$

output $SK'^{(\mathsf{KI})}_{i,j}$;

$\underline{\underline{\mathsf{Sign}_{\mathsf{KI}}(SK_j^{(\mathsf{KI})}, j, M)}}$

$(SKP^{(\mathsf{PS})}_{*\dashrightarrow j}, W) \leftarrow SK_j^{(\mathsf{KI})}$;
$ps \leftarrow \mathsf{PSig}_{\mathsf{PS}}(SKP^{(\mathsf{PS})}_{*\dashrightarrow j}, M, W)$;

output $\langle j, (W, ps)\rangle$;

$\underline{\underline{\mathsf{Vrfy}_{\mathsf{KI}}(PK^{(\mathsf{KI})}, M, \langle j, s\rangle)}}$

$PK_* \leftarrow PK^{(\mathsf{KI})}$;
$(W, ps) \leftarrow s$; $PK^* \leftarrow \mathsf{ID}^{(\mathsf{PS})}_{\mathsf{PS}}(W, ps)$;
`if` $(PK^* \neq (\underbrace{PK_*, \ldots, PK_*}_{j+1}, \cdot))$ `then`
 output 0;
`else`
 output $\mathsf{PVrf}_{\mathsf{PS}}(PK_*, M, W, ps)$;

The following theorem holds for the above construction.

Theorem 1. *Suppose there exists* $(\tau_{\mathsf{KI}}, \epsilon_{\mathsf{KI}}, q^{\mathsf{sig}}_{\mathsf{KI}}, q^{\mathsf{sec}}_{\mathsf{KI}})$*-Adversary* F_{KI} *against* KI *as constructed above with probability* ϵ_{KI}*, with running time* τ_{KI}*,* $q^{\mathsf{sig}}_{\mathsf{KI}}$ *queries to the signing oracle,* $q^{\mathsf{sec}}_{\mathsf{KI}}$ *queries to the key exposure oracle then there exists* $(\tau_{\mathsf{PS}}, \epsilon_{\mathsf{PS}}, q^{\mathsf{sig}}_{\mathsf{PS}}, q^{\mathsf{sec}}_{\mathsf{PS}}, q^{\mathsf{Dlg}}_{\mathsf{PS}})$*-Adversary* F_{PS} *against* PS *with* $\tau_{\mathsf{PS}} = \tau_{\mathsf{KI}}$*,* $\epsilon_{\mathsf{PS}} = \epsilon_{\mathsf{PS}}$*,* $q^{\mathsf{sig}}_{\mathsf{PS}} = q^{\mathsf{sig}}_{\mathsf{KI}}$*,* $q^{\mathsf{sec}}_{\mathsf{PS}} = q^{\mathsf{sec}}_{\mathsf{KI}}$*,* $q^{\mathsf{Dlg}}_{\mathsf{PS}} = q^{\mathsf{sec}}_{\mathsf{KI}}$*.*

Proof. We construct the signing oracle $O^{\mathsf{KI}}_{\mathsf{sig}}$ and the key exposure oracle $O^{(\mathsf{KI})}_{\mathsf{sec}}$ from $O^{(\mathsf{PS})}_{\mathsf{sig}}, O^{(\mathsf{PS})}_{\mathsf{sec}}$ and $O^{(\mathsf{PS})}_{\mathsf{Dlg}}$ as follows.

$\underline{\underline{O^{(\mathsf{KI})}_{\mathsf{sig}}(M, j)}}$

output $O^{(\mathsf{PS})}_{\mathsf{sig}}(\text{“p”}, (\underbrace{*, *, \ldots, *}_{j+1}, j).M, W_j)$;

$\underline{\underline{O^{(\mathsf{KI})}_{\mathsf{sec}}(\texttt{query})}}$

```
if (query = ("s", j)) then
```
$$(SK_j, PK_j) \leftarrow O^{(\mathsf{PS})}_{\mathsf{sec}}(\text{“s”}, j);$$
$$(SKP_{*\dashrightarrow j}, W_{j+1}) \leftarrow \begin{bmatrix} O^{(\mathsf{PS})}_{\mathsf{Dlg}}(\text{“d”}, (\underbrace{*, *, \ldots, *}_{j+1}), W_j, \Lambda), \\ \mathsf{Dlg}_{\mathsf{PPS}}(PK_*, PK_j, SK_j) \end{bmatrix};$$
output $(SKP_{*\dashrightarrow j}, W_{j+1})$;
```
else
```
output $\perp$;

Then $F_{\mathsf{PS}}^{O^{(\mathsf{PS})}_{\mathsf{sig}}, O^{(\mathsf{PS})}_{\mathsf{sec}}, O^{(\mathsf{PS})}_{\mathsf{Dlg}}}(PK^{(\mathsf{PS})}_*) = (M, W, \sigma, PK^{(\mathsf{PS})}_*)$ where $(M, \langle j, (W, \sigma)\rangle) = F_{\mathsf{KI}}^{O^{(\mathsf{KI})}_{\mathsf{sig}}, O^{(\mathsf{KI})}_{\mathsf{sec}}}(PK^{(\mathsf{PS})}_*)$ is the adversary as desired. Since if F_{KI} can forge a valid signature $\langle j, \sigma\rangle$ for the message M then it is easy to see from the construction that $\sigma = (W, ps)$ is also a valid pair of a warrant and a proxy signature for the message M. Further, the scheme Π_{PS} is not $((\underbrace{*, \ldots, *}_{j+1}, j), W, Q^{(\mathsf{PS})})$-proxy-signable where $Q_{\mathsf{PS}} = (Q^{(\mathsf{PS})}_{\mathsf{sig}}, Q^{(\mathsf{PS})}_{\mathsf{sec}} Q^{(\mathsf{PS})}_{\mathsf{Dlg}})$ is a set of valid query to the oracles of Π_{PS} and $(\text{“p”}, (\underbrace{*, \ldots, *}_{j+1}, j), M, W)$ is never queried to $O^{(\mathsf{PS})}_{\mathsf{sig}}$.

Further, the scheme is $(N-1, N)$ KI since if the adversary who gets the signing key of periods $j_1, \ldots, j_{N-1}$ can compute the signature of the period $j_N \notin \{j_1, \ldots, j_{N-1}\}$ then the adversary can compute the proxy signature which is not proxy signable. □

Efficiency: The running time of each algorithm $\mathsf{Gen}_{\mathsf{KI}}, \mathsf{Upd}_{\mathsf{KI}}, \mathsf{Sign}_{\mathsf{KI}}$ and $\mathsf{Vrfy}_{\mathsf{KI}}$ becomes as follows, where $\tau^{(\mathsf{SIG})}_{\mathsf{Alg}}$ denotes the running time of the algorithm Alg for the signature scheme SIG.

$$\tau^{(\mathsf{KI})}_{\mathsf{Gen}} = 2 \cdot \tau^{(\mathsf{PS})}_{\mathsf{Gen}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{P}}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{D}}}, \qquad \tau^{(\mathsf{KI})}_{\mathsf{Upd}^*} = (N+1) \cdot \left(\tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{D}}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{D}}}\right) + \tau^{(\mathsf{PS})}_{\mathsf{Gen}},$$
$$\tau^{(\mathsf{KI})}_{\mathsf{Upd}} = \mathcal{O}(1), \qquad \tau^{(\mathsf{KI})}_{\mathsf{Sign}} = \tau^{(\mathsf{PS})}_{\mathsf{PSig}}, \qquad \tau^{(\mathsf{KI})}_{\mathsf{Vrfy}} = \tau^{(\mathsf{PS})}_{\mathsf{PVrf}} + \tau^{(\mathsf{PS})}_{\mathsf{ID}}$$

3.2 KI → Proxy

PS with n designators can be constructed constructed from $(c \cdot n - 1, c \cdot n)$ KI as follows (where c is the total number of self delegation allowed for each delegator.) In key generation phase, c signer keys $SK_{j \cdot c}, SK_{j \cdot c+1}, \ldots, SK_{(j-1)\cdot c-1}$ is assigned to designator j. the signer key $SK_{j \cdot c}$ is used for (ordinary) signing, proxy signing and delegation. The other key is used for self proxy signing and self delegation.

Delegation is simply based on so-called “certificate chain”. That is, to delegate the signing right of user i to user j, the user i simply compute the *warrant* containing information of the public key of user i, the limitation of the delegation and the signature of user j. In our construction the warrant W is of the form $W = (W', \omega, \mathsf{Sign}_{\mathsf{KI}}(SK, (W', \omega)))$ where W' is the warrant of previous delegation and $\omega = (l_1, l_2, \mathtt{usage})$ describes the limitation of the current delegation, namely, l_1 and l_2 denote the range of possible secret keys used for self proxy signing (therefore, l_1, l_2 only make sense in the self delegation.) This type of warrant prevents the user i with warrants $W_1, \ldots, W_n$ from computing a valid proxy signature of any warrant other than $W_1, \ldots, W_n$.

Note that different signer key of KI is assigned for each self delegation. This prevents the attacker who gets a signer key which can be used with some self delegation from computing a valid proxy signature for the other self delegation. The concrete security reduction can be shown by the following theorem.

Theorem 2. *It is possible to construct* PS *(with n designators and the total number of self delegation allowed for each delegator is less than a constant c) from* $(c \cdot n - 1, c \cdot n)$ KI *in such a way that if there exists* $(\tau_{\mathsf{PS}}, \epsilon_{\mathsf{PS}}, q_{\mathsf{PS}}^{\mathsf{sig}}, q_{\mathsf{PS}}^{\mathsf{sec}}, q_{\mathsf{PS}}^{\mathsf{Dlg}})$*-Adversary* F_{PS} *against* PS *then there exists* $(\tau_{\mathsf{KI}}, \epsilon_{\mathsf{KI}}, q_{\mathsf{KI}}^{\mathsf{sig}}, q_{\mathsf{KI}}^{\mathsf{sec}})$*-Adversary* F_{KI} *against* KI *with* $\tau_{\mathsf{KI}} = \tau_{\mathsf{PS}}$, $\epsilon_{\mathsf{KI}} = \epsilon_{\mathsf{PS}}$, $q_{\mathsf{KI}}^{\mathsf{sig}} = q_{\mathsf{PS}}^{\mathsf{sig}} + q_{\mathsf{PS}}^{\mathsf{Dlg}}$ *and* $q_{\mathsf{KI}}^{\mathsf{sec}} \leq q_{\mathsf{PS}}^{\mathsf{sec}} + c \cdot q_{\mathsf{PS}}^{\mathsf{Dlg}}$

The proof is given in the full version [13].

EFFICIENCY: The running time of $\mathsf{Gen}_{\mathsf{PS}}, \mathsf{Sign}_{\mathsf{PS}}, \mathsf{Vrfy}_{\mathsf{PS}}, \mathsf{Dlg}_{\mathsf{DPS}}, \mathsf{Dlg}_{\mathsf{PPS}}, \mathsf{PSig}_{\mathsf{PS}}$ and $\mathsf{PVrf}_{\mathsf{PS}}$ in the construction of the above theorem, become as follows where L denotes the length of the delegation chain.

$$\tau_{\mathsf{Gen}}^{(\mathsf{PS})} = \tau_{\mathsf{Gen}}^{(\mathsf{KI})} + c\left(\tau_{\mathsf{Upd}^*}^{(\mathsf{KI})} + \tau_{\mathsf{Upd}}^{(\mathsf{KI})}\right), \qquad \tau_{\mathsf{Sign}}^{(\mathsf{PS})} = \tau_{\mathsf{Sign}}^{(\mathsf{KI})}, \qquad \tau_{\mathsf{Vrfy}}^{(\mathsf{PS})} = \tau_{\mathsf{Vrfy}}^{(\mathsf{KI})},$$
$$\tau_{\mathsf{Dlg_D}}^{(\mathsf{PS})} = \tau_{\mathsf{Sign}}^{(\mathsf{KI})}, \qquad \tau_{\mathsf{Dlg_P}}^{(\mathsf{PS})} = \mathcal{O}(1),$$
$$\tau_{\mathsf{PSig}}^{(\mathsf{PS})} = \tau_{\mathsf{Sign}}^{(\mathsf{KI})}, \qquad \tau_{\mathsf{PVrf}}^{(\mathsf{PS})} = L \cdot \tau_{\mathsf{Vrfy}}^{(\mathsf{KI})}, \qquad \tau_{\mathsf{ID}}^{(\mathsf{PS})} = \mathcal{O}(L)$$

4 The Hierarchy of Key Evolving Signatures

In this section we show the hierarchy among the key evolving signatures. Namely, we show that intrusion-resilient signatures imply $(N-1, N)$ key-insulated signatures and vice versa, and that proxy signatures (and thus $(N-1, N)$ key-insulated signatures) imply forward-secure signatures. The results are summarized below, each followed by a brief overview of the proof. In some cases the complete formal constructions and proofs are omitted from this extended abstract, and can be found in the full version of our paper [13].

Theorem 3 (IR $\rightarrow$ KI). *It is possible to construct* KI *from* IR *in such a way that if there exists* $(\tau_{\mathsf{KI}}, \epsilon_{\mathsf{KI}}, q_{\mathsf{KI}}^{\mathsf{sig}}, q_{\mathsf{KI}}^{\mathsf{sec}})$*-Adversary* F_{KI} *which breaks* KI *then there exists* $(\tau_{\mathsf{IR}}, \epsilon_{\mathsf{IR}}, q_{\mathsf{IR}}^{\mathsf{sig}}, q_{\mathsf{IR}}^{\mathsf{sec}})$*-Adversary* F_{IR} *which breaks* IR *with* $\tau_{\mathsf{IR}} = \tau_{\mathsf{KI}}$, $\epsilon_{\mathsf{IR}} = \epsilon_{\mathsf{KI}}$, $q_{\mathsf{IR}}^{\mathsf{sig}} = q_{\mathsf{KI}}^{\mathsf{sig}}$ *and* $q_{\mathsf{IR}}^{\mathsf{sec}} = q_{\mathsf{KI}}^{\mathsf{sec}}$.

The reduction is based on the following idea: all the initial data of IR is stored in the base of KI and the signer of the KI only stores signer key of the current period. Then the random access to the key is possible by simply computing the signer key of any period from the initial state. The formal details are given below.

Proof. We construct $(N-1,N)$ key-insulated signature $\Pi_{\mathsf{KI}} = (\mathsf{Gen}_{\mathsf{KI}}, \mathsf{Upd}^*_{\mathsf{KI}}, \mathsf{Upd}_{\mathsf{KI}}, \mathsf{Sign}_{\mathsf{KI}}, \mathsf{Vrfy}_{\mathsf{KI}})$ from intrusion-resilient signature $\Pi_{\mathsf{IR}} = (\mathsf{Gen}_{\mathsf{IR}}, \mathsf{Upd}^*_{\mathsf{IR}}, \mathsf{Upd}_{\mathsf{IR}}, \mathsf{Refr}^*_{\mathsf{IR}}, \mathsf{Refr}_{\mathsf{IR}}, \mathsf{Sign}_{\mathsf{IR}}, \mathsf{Vrfy}_{\mathsf{IR}})$ as follows.

$\underline{\underline{\mathsf{Gen}_{\mathsf{KI}}(1^k, N)}}$

$(SKB^{(\mathsf{IR})}_{0.0}, SKS^{(\mathsf{IR})}_{0.0}, PK^{(\mathsf{IR})}) \leftarrow \mathsf{Gen}_{\mathsf{IR}}(1^k, N)$;
$SK^{*(\mathsf{KI})} \leftarrow (SKS^{(\mathsf{IR})}_{0.0}, SKB^{(\mathsf{IR})}_{0.0})$; $\quad SK^{(\mathsf{KI})}_0 \leftarrow SKS^{(\mathsf{IR})}_{0.0}$; $\quad PK^{(\mathsf{KI})} \leftarrow PK^{(\mathsf{IR})}$;

output $(SK^{*(\mathsf{KI})}, SK^{(\mathsf{KI})}_0, PK^{(\mathsf{KI})})$;

$\underline{\underline{\mathsf{Upd}^*_{\mathsf{KI}}(SK^{*(\mathsf{KI})}, i, j)}}$

$(SKB, SKS) \leftarrow SK^{*(\mathsf{KI})}$;
`for` $n = 0$ `to` $j-1$ `do`
  $(SKB, SKU) \leftarrow \mathsf{Upd}^*_{\mathsf{IR}}(SKB)$;
  $SKS \leftarrow \mathsf{Upd}_{\mathsf{IR}}(SKS, SKU)$;
  $(SKB, SKR) \leftarrow \mathsf{Refr}^*_{\mathsf{IR}}(SKB)$;
  $SKS \leftarrow \mathsf{Refr}_{\mathsf{IR}}(SKS, SKR)$;
$SK'^{(\mathsf{KI})}_{i,j} \leftarrow SKS$;

output $SK'^{(\mathsf{KI})}_{i,j}$;

$\underline{\underline{\mathsf{Upd}_{\mathsf{KI}}(SK^{(\mathsf{KI})}_i, SK'^{(\mathsf{KI})}_{i,j})}}$

$SK^{(\mathsf{KI})}_j \leftarrow SK'^{(\mathsf{KI})}_{i,j}$;

output $SK^{(\mathsf{KI})}_j$;

$\underline{\underline{\mathsf{Sign}_{\mathsf{KI}}(SK^{(\mathsf{KI})}_j, j, M)}}$

output $\mathsf{Sign}_{\mathsf{IR}}(SK^{(\mathsf{KI})}_j, j, M)$;

$\underline{\underline{\mathsf{Vrfy}_{\mathsf{KI}}(PK^{(\mathsf{KI})}, M, \langle j, s\rangle)}}$

output $\mathsf{Vrfy}_{\mathsf{IR}}(PK^{(\mathsf{KI})}, M, \langle j, s\rangle)$;

We also construct the signing oracle $O^{(\mathsf{KI})}_{\mathsf{sig}}$ and the key exposure oracle $O^{(\mathsf{KI})}_{\mathsf{sec}}$ of KI from $O^{(\mathsf{IR})}_{\mathsf{sig}}$ and $O^{(\mathsf{IR})}_{\mathsf{sec}}$ as follows.

$\underline{\underline{O^{(\mathsf{KI})}_{\mathsf{sig}}(M, j)}}$

output $O^{(\mathsf{IR})}_{\mathsf{sig}}(M, j.1)$;

$\underline{\underline{O^{(\mathsf{KI})}_{\mathsf{sec}}(\mathtt{query})}}$

`if (query =` (“s”, j)`) then`
  output $O^{(\mathsf{IR})}_{\mathsf{sec}}$(“s”, $j.1$);
`else`
  output $\perp$;

Then $F^{O^{(\mathsf{IR})}_{\mathsf{sig}}, O^{(\mathsf{IR})}_{\mathsf{sec}}}_{\mathsf{IR}}(PK^{(\mathsf{IR})}) = F^{O^{(\mathsf{KI})}_{\mathsf{sig}}, O^{(\mathsf{KI})}_{\mathsf{sec}}}_{\mathsf{KI}}(PK^{(\mathsf{IR})})$ is the adversary as desired. This is because KI and two oracles for KI are constructed in such a way that $SK^{(\mathsf{KI})}_j = SK^{(\mathsf{IR})}_{j.1}$ holds and the signing algorithm and the verification algorithm are exactly the same as those of IR. Therefor, if F_{KI} can produce a valid signature $(M, \langle j, sig\rangle)$

such that the scheme is not (j, Q^{KI})-compromised and (M, j) is never queried to $O^{\mathsf{KI}}_{\mathsf{sig}}$ then $\langle j, sig\rangle$ is also valid in IR and the scheme is not (j, Q^{IR})-compromised and $(M, j.1)$ is never queried to $O^{\mathsf{IR}}_{\mathsf{sig}}$. Further, the resulting KI is $(N-1, N)$ KI since the key exposure of $N-1$ point in KI is corresponding to the key exposure of $N-1$ signer secret key of IR and no base key of IR is compromised. Therefore the security of the remaining signing key can be guaranteed by the IR property. □

We note that this construction is in fact a consolidation of earlier proofs we got regarding intermediate constructions, namely showing IR implies KI-FS and KI-FS implies KI. This intermediate notion of KI-FS is defined, and the corresponding reductions are proved, in the full version of our paper [13].

EFFICIENCY: The running time of $\mathsf{Gen}_{\mathsf{KI}}, \mathsf{Upd}^*_{\mathsf{KI}}, \mathsf{Upd}_{\mathsf{KI}}, \mathsf{Sign}_{\mathsf{KI}}$ and $\mathsf{Vrfy}_{\mathsf{KI}}$ in the above construction become as follows.

$$\tau^{(\mathsf{KI})}_{\mathsf{Gen}} = \tau^{(\mathsf{IR})}_{\mathsf{Gen}}, \qquad \tau^{(\mathsf{KI})}_{\mathsf{Upd}^*} = N \cdot \left(\tau^{(\mathsf{IR})}_{\mathsf{Upd}^*} + \tau^{(\mathsf{IR})}_{\mathsf{Upd}} + \tau^{(\mathsf{IR})}_{\mathsf{Refr}^*} + \tau^{(\mathsf{IR})}_{\mathsf{Refr}}\right),$$
$$\tau^{(\mathsf{KI})}_{\mathsf{Upd}} = \mathcal{O}(1), \qquad \tau^{(\mathsf{KI})}_{\mathsf{Sign}} = \tau^{(\mathsf{IR})}_{\mathsf{Sign}}, \qquad \tau^{(\mathsf{KI})}_{\mathsf{Vrfy}} = \tau^{(\mathsf{IR})}_{\mathsf{Vrfy}}.$$

Theorem 4 ($\mathsf{KI} \to \mathsf{IR}$). *It is possible to construct* IR *from* $(N-1, N)$ KI *in such a way that if there exists* $(\tau_{\mathsf{IR}}, \epsilon_{\mathsf{IR}}, q^{\mathsf{sig}}_{\mathsf{IR}}, q^{\mathsf{sec}}_{\mathsf{IR}})$*-Adversary* F_{IR} *which breaks* IR *then there exists* $(\tau_{\mathsf{IR}}, \epsilon_{\mathsf{KI}}, q^{\mathsf{sig}}_{\mathsf{KI}}, q^{\mathsf{sec}}_{\mathsf{KI}})$*-Adversary* F_{IR} *which breaks* KI *with* $\tau_{\mathsf{KI}} = \tau_{\mathsf{IR}}$*,* $\epsilon_{\mathsf{KI}} = \epsilon_{\mathsf{IR}}$*,* $q^{\mathsf{sig}}_{\mathsf{KI}} = q^{\mathsf{sig}}_{\mathsf{IR}}$ *and* $q^{\mathsf{sec}}_{\mathsf{KI}} = q^{\mathsf{sec}}_{\mathsf{IR}}$*.*

The reduction is constructed as follows. In key generation phase the key generation algorithm of KI outputs the secret keys $SK_0, \ldots, SK_N$ of *all* the time periods. Then $(SK_0, SK_1 \oplus R_1, SK_2 \oplus R_2, \ldots, SK_N \oplus R_N)$ is given to the signer as the signing key SKS and $(R_1, R_2, \ldots, R_N)$ is given to the base as its base key SKB where $R_1, R_2, \ldots, R_N$ are random data. SKS and SKB for time period j are of the form $(SK_j, SK_{j+1} \oplus R_{j+1}, SK_{j+2} \oplus R_{j+2}, SK_N \oplus R_N)$ and $(R_{j+1}, R_{j+2}, \ldots, R_N)$, respectively and the signature for the message M in the time period j is simply computed by $\mathsf{Sign}_{\mathsf{KI}}(SK_j, M)$. Further, random date R_is are updated by the refresh algorithms. By this simple construction we can construct IR since

- The adversary knows only the secret key of the time period j if the adversary can successfully attack the signer in the time period j. Further, the knowledge of the signing key of the time period j does not help to forge the signature for the other time period.
- The adversary knows no information about the signing key of any period even if the adversary successfully attack the base.
- The adversary knows no information about the past key even if the adversary successfully attack the signer and the base in the same time period.

Theorem 5 ($\mathsf{PS} \to \mathsf{FS}$). *It is possible to construct* FS *from* PS *in such a way that if there exists* $(\tau_{\mathsf{FS}}, \epsilon_{\mathsf{FS}}, q^{\mathsf{sig}}_{\mathsf{FS}}, q^{\mathsf{sec}}_{\mathsf{FS}})$*-Adversary* F_{FS} *against* FS *then there exists* $(\tau_{\mathsf{PS}}, \epsilon_{\mathsf{PS}}, q^{\mathsf{sig}}_{\mathsf{PS}}, q^{\mathsf{sec}}_{\mathsf{PS}}, q^{\mathsf{Dlg}}_{\mathsf{PS}})$*-Adversary* F_{PS} *against* PS *with* $\tau_{\mathsf{PS}} = \tau_{\mathsf{FS}}$*,* $\epsilon_{\mathsf{PS}} = \epsilon_{\mathsf{PS}}$*,* $q^{\mathsf{sig}}_{\mathsf{PS}} = q^{\mathsf{sig}}_{\mathsf{FS}}$*,* $q^{\mathsf{sec}}_{\mathsf{PS}} = q^{\mathsf{sec}}_{\mathsf{FS}}(=1)$*,* $q^{\mathsf{Dlg}}_{\mathsf{PS}} = q^{\mathsf{sec}}_{\mathsf{FS}}(=1)$*.*

The reduction is constructed in such a way that the signing key of the time period j corresponds to the self-delegation key of delegation level $j+1$. Though this is a simple construction, forward-security can be achieved since an attacker is not able to get the signing key of lower delegation level even if the attacker gets the self delegation key of some delegation level.

EFFICIENCY: The running time of $\mathsf{Gen}_{\mathsf{FS}}, \mathsf{Upd}_{\mathsf{FS}}, \mathsf{Sign}_{\mathsf{FS}}$ and $\mathsf{Vrfy}_{\mathsf{FS}}$ in the above construction become as follows.

$$\tau^{(\mathsf{FS})}_{\mathsf{Gen}} = \tau^{(\mathsf{PS})}_{\mathsf{Gen}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{D}}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{P}}}, \qquad \tau^{(\mathsf{FS})}_{\mathsf{Upd}} = \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{D}}} + \tau^{(\mathsf{PS})}_{\mathsf{Dlg}_{\mathsf{P}}},$$
$$\tau^{(\mathsf{FS})}_{\mathsf{Sign}} = \tau^{(\mathsf{PS})}_{\mathsf{PSig}}, \qquad \tau^{(\mathsf{FS})}_{\mathsf{Vrfy}} = \tau^{(\mathsf{PS})}_{\mathsf{PVrf}} + \tau^{(\mathsf{PS})}_{\mathsf{ID}}$$

The following corollary is immediate from Theorem 2 and Theorem 5.

Corollary 1 ($\mathsf{KI} \to \mathsf{FS}$)**.** *It is possible to construct* FS *from* KI *in such a way that if there exists* $(\tau_{\mathsf{FS}}, \epsilon_{\mathsf{FS}}, q^{\mathsf{sig}}_{\mathsf{FS}}, q^{\mathsf{sec}}_{\mathsf{FS}})$*-Adversary* F_{FS} *against* FS *then there exists* $(\tau_{\mathsf{KI}}, \epsilon_{\mathsf{KI}}, q^{\mathsf{KI}}_{\mathsf{KI}}, q^{\mathsf{KI}}_{\mathsf{KI}})$*-Adversary* F_{KI} *against* KI *with* $\tau_{\mathsf{KI}} = \tau_{\mathsf{FS}}$*,* $\epsilon_{\mathsf{KI}} = \epsilon_{\mathsf{FS}}$*,* $q^{\mathsf{sig}}_{\mathsf{KI}} = q^{\mathsf{sig}}_{\mathsf{FS}} + q^{\mathsf{sec}}_{\mathsf{FS}}$ *and* $q^{\mathsf{sec}}_{\mathsf{KI}} = N \cdot q^{\mathsf{sec}}_{\mathsf{FS}}$*.*

References

1. R. Anderson, *Two remarks on public key cryptology*, available at `http://www.cl.cam.ac.uk/users/rja14/`, 2001.
2. M. Bellare and S. Miner, *A forward-secure digital signature scheme*, Proc. Crypto'92, Lecture Notes in Computer Science, vol. 1666, pp. 15–19, 1999.
3. A. Boldyreva, A. Palacio and B. Warinschi, *Secure Proxy Signature Scheme for Delegation of Signing Rights*, IACR ePrint Archive, available at `http://eprint.iacr.org/2003/096/`, 2003.
4. Y. Dodis, J. Katz, S. Xu and M. Yung, *Key-Insulated Public Key Cryptosystems*, Proc. Eurocrypt 2002, Lecture Notes in Computer, vol. 2332, pp. 65–82, 2002.
5. Y. Dodis, J. Katz, S. Xu and M. Yung, *Strong Key-Insulated Signature Schemes*, Proc. PKC2003, Lecture Notes in Computer Science, vol. 2567, pp. 130–144, 2003.
6. M. Gassr, A. Goldstein, C. Kaufman and B. Lampson, *The Digital Distributed Security Architecture*, Proc. National Computer Security Conference, 1989.
7. G. Itkis, *Intrusion-Resilient Signatures: Generic Constructions, or Defeating Strong Adversary with Minimal Assumptions*, Proc. SCN2002, Lecture Notes in Computer Science, vol. 2576, pp. 102–118, 2002.
8. G. Itkis and L. Reyzin, *Forward-secure signatures with optimal signing and verifying*, Proc. Crypto2001, Lecture Notes in Computer Science, vol. 2139, pp. 332-354, 2001.
9. G. Itkis and L. Reyzin, *SiBIR: Signer-Base Intrusion-Resilient Signatures*, Proc. Crypto2002, Lecture Notes in Computer Science, vol. 2442, pp. 499–514, 2002.
10. A. Kozlov and L. Reyzin, *Forward-Secure Signatures with Fast Key Update*, Proc. SNC2002, Lecture Notes in Computer Science, vol. 2576, pp. 241–256, 2002.
11. H. Krawczyk, *Simple forward-secure signatures from any signature scheme.*, Proc. the 7th ACM Conference on Computer and Communications Security, pp. 108–115, 2000.

12. T. Malkin, D. Micciancio and S. Miner, *Efficient generic forward-secure signatures with an unbounded number of time periods*, Proc. Eurocrypt2002, Lecture Notes in Computer Science, vol. 2332, pp. 400–417, 2002.
13. T. Malkin, S. Obana and M. Yung, *The Hierarchy of Key Evolving Signatures and a Characterization of Proxy Signatures*, full version of this paper available at the IACR ePrint Archive, `http://eprint.iacr.org/2004/052`, 2004.
14. M. Mambo, K. Usuda and E. Okamoto, *Proxy signatures for delegating signing operation*, Proc. the 3rd ACM Conference on Computer and Communications Security, pp. 48–57, 1996.
15. M. Naor and M. Yung, *Universal One-Way Hash Functions and their Cryptographic Applications*, In Proceedings of the ACM Symposium on Theory of Computing, 1989, pp. 33–43..
16. J. Rompel, *One-way functions are necessary and sufficient for secure signatures.* In Proceedings of the ACM Symposium on Theory of Computing, 1990, pp. 387–394.