A Note On Game-Hopping Proofs

Alexander W. Dent

Royal Holloway, University of London
Egham Hill, Egham, Surrey, TW20 0EX, U.K.
a.dent@rhul.ac.uk

Abstract. Game hopping is a method for proving the security of a cryptographic scheme. In a game hopping proof, we observe that an attacker running in a particular attack environment has an unknown probability of success. We then slowly alter the attack environment until the attackers success probability can be computed. We also bound the increase in the attacker's success probability caused by the changes to the attack environment. Thus, we can deduce a bound for the attacker's success probability in the original environment. Currently, there are three known "types" of game hop: transitions based on indistinguishability, transitions based on failure events, and bridging steps. This note introduces a fourth type of game hop.

1 Introduction

Some confidence about the security of a cryptographic scheme can be obtained if a proof of security for the scheme is exhibited. A proof of security demonstrates that, if the scheme can be broken in some formal security model, then some computationally hard problem can be broken. One method for proving the security of a scheme is to use a game hopping proof [1, 2]. In a game hopping proof, we observe that an attacker running in a particular attack environment has an unknown probability of success. We then slowly alter the attack environment until the attacker's success probability can be computed. We also bound the increase in the attacker's success probability caused by the changes to the attack environment. Thus, we can deduce a bound for the attacker's success probability in the original environment.

For our purposes, we will consider a probabilistic, polynomial-time attacker $\mathcal{A}$. The attacker will run in one of two environments (Game 1 and Game 2) that are parameterised by a security parameter λ. The environment which will provide the attacker with initial inputs and, perhaps, access to certain oracles. The attacker will terminate with some final output, which will then be assessed to see if the attacker "won". The event that the attacker wins Game i will be denoted S_i. In this paper, we are concerned with techniques that relate $Pr[S_1]$ and $Pr[S_2]$. We are particularly interested in the case where $Pr[S_1]$ is negligible as a function of the security parameter λ if and only if $Pr[S_2]$ is negligible as a function of the security parameter λ. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is defined to be negligible if, for all $c \in \mathbb{Z}$, there exists k_0 such that $|f(k)| \leq k^{-c}$ for all $k \geq k_0$.

There are currently types of game hop:

Bridging steps This type of game hop merely rephrases the environment and (from the attacker's point of view) no part of the environment actually changes. Hence, the attacker's probability of breaking the scheme remains the same.

Transitions based on indistinguishability In this type of game hop, an input that the attacker receives (either at the start of their execution or as a response from an oracle) is changed. Instead of drawing that value from the correct distribution, as specified in the original attack environment, the value is drawn from a second distribution that is computationally indistinguishable from the correct one. If the attacker's behaviour significantly alters as a result of this game hop, then we may distinguish the two distributions. Hence, since the probability that any algorithm distinguishes the two distributions is negligible, the attacker's success probability may only increase by a negligible amount.

Transitions based on (small) failure events In this type of game hop, the two environments interact with the attacker in exactly the same way unless a certain "error" event E occurs. It is assumed that this event occurs with negligible probability. The fact that an attacker's success probability only increases by a negligible amount when moving between these games is a result of the following lemma:

Lemma 1 (Difference Lemma). *Let E be some "error event" such that $S_1|\neg E$ occurs if and only if $S_2|\neg E$ occurs. Then*

$$|Pr[S_1] - Pr[S_2]| \leq Pr[E]$$

Our contribution In this short note, we will describe a new type of game hop, which we term a *transition based on large failure events*. Here, just as in a transition based on small failure events, we assume that two environments interact with each other in exactly the same way unless a certain error event E occurs. However, now we assume that the probability that this error occurs is very large, but not totally overwhelming. Furthermore, we assume that the event E is independent of the attacker's success probability S_1.

2 Transitions based on large failure events

We distinguish between two different types of game with which an attacker may be interacting: a game in which the attacker is required to directly compute an answer (such as occurs when an attacker is trying to forge a digital signature) and a game in which the attacker is attempting to distinguish between two possible situations (such as occurs when an attacker is trying to discover some information about an encrypted message).

2.1 Direct Computation

We begin by considering the case in which an attacker is attempting to compute some value (for example, computing a forgery for a signature scheme). If we are to use game hopping techniques in such a situation, we need to directly relate $Pr[S_1]$ to $Pr[S_2]$.

Let E be an event that can occur during the execution of the attacker in Game 1. We assume that $\neg E$ is non-negligible, that the environment can recognise when E occurs, and that the events S_1 and E are independent, i.e. that $Pr[S_1 \wedge E] = Pr[S_1] \cdot Pr[E]$. Let Game 2 be the attack environment which is identical to Game 1 until the moment that E occurs, at which point the environment halts the attack game. In such a situation, the attacker is considered not to have succeeded; if E does not occur then the attacker wins in Game 2 if and only if it would win in Game 1. In this situation, it is easy to see that

$$\begin{aligned} Pr[S_2] &= Pr[S_1 \wedge \neg E] \\ &= Pr[S_1] \cdot Pr[\neg E] \,. \end{aligned}$$

In particular, note that $Pr[S_1]$ is non-negligible if and only if $Pr[S_2]$ is non-negligible.

It may be thought that this situation rarely occurs in practice, but it is actually quite common in cryptographic proofs. For example, consider the case in which we are trying to prove the security of a digital signature scheme against an attacker $\mathcal{A}$ which can obtain the public keys of several individuals. This may be represented in Game 1 as the attacker being initially given q public key values. The attacker is deemed to succeed if it outputs a forgery for a public key that it has been given. In Game 2, however, the attack environment randomly chooses a critical number $i \in \{1, 2, \ldots, q\}$. Game 2 runs exactly the same way as Game 1, but the attacker is deemed to have won only if it outputs a valid forgery for the i-th public key that it was given. We may relate these games by defining E to be the event that $\mathcal{A}$ outputs an attempted forgery for a public key not equal to the i-th public key. Note that E is independent of S_1, that E is recognisable by the environment, and that $Pr[\neg E] = 1/q$. Hence, the use of the above observation immediately gives us that

$$Pr[S_2] = Pr[S_1]/q \,.$$

If the attacker is polynomially bounded, then q must be polynomially bounded and so $P[S_1]$ is negligible if and only if $Pr[S_1]$ is negligible.

2.2 Indistinguishability

The second case to consider is that of an attacker trying to solve a decision problem. In such cases, the attacker is attempting to discern a hidden bit b. The attacker is assumed to output a guess b' for b in all of its executions, and is thought to have "won" if $b' = b$. The scheme is thought to be secure if $|Pr[S_1] - 1/2|$ is small (negligible). This value is known as the attacker's *advantage*. Hence, unlike the case of direct computation, we are not interested in showing that $Pr[S_1]$ is negligible if and only if $Pr[S_2]$ is negligible. We are interested in showing that $|Pr[S_1] - 1/2|$ is negligible if and only if $|Pr[S_2] - 1/2|$ is negligible. This is situation is, again, quite common in cryptographic proofs.

Let E be an event that can occur during the execution of the attacker in Game 1. We assume that $\neg E$ is non-negligible, that the environment can recognise when E occurs, and that S_1 and E are independent, i.e. $Pr[S_1|E] = Pr[S_1]$. Let Game 2 be the attack environment which is identical to Game 1 until the moment that E occurs, at which point we assume that the environment simulates $\mathcal{A}$'s output by randomly choosing a bit b'. If E does not occur, then the attacker will output the same bit that it did in Game 1. This means that $Pr[S_2|E] = 1/2$ and that $Pr[S_2|\neg E] = Pr[S_1|\neg E] = Pr[S_1]$. It is now easy to see that:

$$\begin{aligned}
|Pr[S_2] - 1/2| &= |Pr[S_2|E]Pr[E] + Pr[S_2|\neg E]Pr[\neg E] - 1/2| \\
&= |Pr[E]/2 + Pr[S_1|\neg E]Pr[\neg E] - 1/2| \\
&= |(1 - Pr[\neg E])/2 + Pr[S_1]Pr[\neg E] - 1/2| \\
&= Pr[\neg E] \cdot |Pr[S_1] - 1/2| \,.
\end{aligned}$$

Hence, the attacker has non-negligible advantage in Game 1 if and only if it has non-negligible advantage in Game 2.

References

1. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advances in Cryptology – Eurocrypt 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer-Verlag, 2006.
2. V. Shoup. Sequences of games: A tool for taming complexity in security proofs. Available from `http://eprint.iacr.org/2004/332/`, 2004.